

# Angular and Linear Velocity Estimation for a Re-Entry Vehicle Using Six Distributed Accelerometers: Theory, Simulation and Feasibility

*Grace A. Clark*

**April 28, 2003**

**U.S. Department of Energy**

Lawrence  
Livermore  
National  
Laboratory

## DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U. S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.

This report has been reproduced directly from the best available copy.

Available electronically at <http://www.doc.gov/bridge>

Available for a processing fee to U.S. Department of Energy  
And its contractors in paper from  
U.S. Department of Energy  
Office of Scientific and Technical Information  
P.O. Box 62  
Oak Ridge, TN 37831-0062  
Telephone: (865) 576-8401  
Facsimile: (865) 576-5728  
E-mail: [reports@adonis.osti.gov](mailto:reports@adonis.osti.gov)

Available for the sale to the public from  
U.S. Department of Commerce  
National Technical Information Service  
5285 Port Royal Road  
Springfield, VA 22161  
Telephone: (800) 553-6847  
Facsimile: (703) 605-6900  
E-mail: [orders@ntis.fedworld.gov](mailto:orders@ntis.fedworld.gov)  
Online ordering: <http://www.ntis.gov/ordering.htm>

OR

Lawrence Livermore National Laboratory  
Technical Information Department's Digital Library  
<http://www.llnl.gov/tid/Library.html>

**Angular and Linear Velocity Estimation  
for a Re-Entry Vehicle Using Six  
Distributed Accelerometers:  
Theory, Simulation and Feasibility**

Grace A. Clark  
Lawrence Livermore National Laboratory  
7000 East Ave., L-130, Livermore, CA 94550  
(925) 423-9759 (Office), (925) 422-2495 (FAX),  
clark9@llnl.gov

April 28, 2003

UCRL-ID-153253

## Acknowledgments

The author gratefully acknowledges the important contributions to this work made by her colleagues. Dr. David H. Chambers has contributed greatly in the area of kinematics and the numerical solution of nonlinear differential equations. In addition, he has been an outstanding source of technical knowledge for the entire project. Ronald J. Kane, the program leader has provided inspired vision and technical leadership for the project. Carlos A. Avalle and his team expertly designed and implemented the DAIMU assembly, electronics and measurement scheme.

## Abstract

This report describes a feasibility study. We are interested in calculating the angular and linear velocities of a re-entry vehicle using six acceleration signals from a distributed accelerometer inertial measurement unit (DAIMU) [8, 10, 11, 9, 12, 13, 14]. Earlier work [12] showed that angular and linear velocity calculation using classic nonlinear ordinary differential equation (ODE) solvers is not practically feasible, due to mathematical and numerical difficulties. This report demonstrates the *theoretical feasibility* of using model-based nonlinear state estimation techniques to obtain the angular and linear velocities in this problem. Practical numerical and calibration issues require additional work to resolve.

We show that the six accelerometers in the DAIMU are not sufficient to provide observability, so additional measurements of the system states are required (e.g. from a Global Positioning System (GPS) unit). Given the constraint that our system cannot use GPS, we propose using the existing on-board 3-axis magnetometer to measure angular velocity. We further show that the six nonlinear ODE's for the vehicle kinematics can be decoupled into three ODE's in the angular velocity and three ODE's in the linear velocity. This allows us to formulate a three-state Gauss-Markov system model for the angular velocities, using the magnetometer signals in the measurement model. This re-formulated model is observable, allowing us to build an Extended Kalman Filter (EKF) for estimating the angular velocities. Given the angular velocity estimates from the EKF, the three ODE's for the linear velocity become algebraic, and the linear velocity can be calculated by numerical integration. Thus, we do not need direct measurements of the linear velocity to provide observability, and the technique is mathematically feasible.

Using a simulation example, we show that the estimator adds value over the numerical ODE solver in the presence of measurement noise. Calculating the velocities in the presence of significant measurement noise is not feasible with a classic ODE solver. The EKF is able to deal effectively with the noise and provide useful angular velocity estimates. The linear velocity estimates for this simulation show numerical difficulties associated with the nonlinear ODE's and the quadrature operation.

Future work will focus on dealing with practical numerical issues and the issue of calibrating the DAIMU to deal with uncertainties in the accelerometer positions and locations.

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Gyroscope vs. Accelerometer Array . . . . .	5
1.2	The DAIMU in a Re-Entry Vehicle . . . . .	9
1.3	Theoretical Analysis . . . . .	9
1.4	Estimator Development and Testing . . . . .	10
<b>2</b>	<b>Kinematic Equations for the Re-Entry Vehicle</b>	<b>11</b>
<b>3</b>	<b>Development of the Kinematic System Equations in State Space Form</b>	<b>16</b>
3.1	The Vector ODE Relating $\underline{A}(t)$ and $\underline{\dot{B}}(t)$ . . . . .	17
3.1.1	The Configuration Matrix $\tilde{T}$ . . . . .	17
3.1.2	The Centripetal Acceleration Vector $\underline{C}[\underline{B}(t)]$ . . . . .	19
3.1.3	Decoupling the Continuous-Time Kinematic Equations . . . . .	20
3.2	Formulating the Kinematic Equations as an Observable Continuous-Time Gauss-Markov System Model . . . . .	21
3.2.1	System Plant . . . . .	22
3.2.2	System Measurements . . . . .	22
3.2.3	Estimation of the Linear Velocities . . . . .	22
3.3	Discrete-Time Gauss-Markov Model . . . . .	23
3.3.1	Discretization Using a 2nd-Order Semi-Implicit Method . . . . .	23
3.3.2	Discretized System Plant . . . . .	24

3.3.3	Discrete-Time System Output Measurements: . . . . .	25
3.3.4	Estimation of the Linear Velocities $\underline{v}(k)$ . . . . .	25
3.3.5	Jacobian Matrices Required by the Extended Kalman Filter	26
<b>4</b>	<b>The Extended Kalman Filter Algorithm</b>	<b>27</b>
4.1	Discrete-Time Nonlinear Gauss-Markov Model . . . . .	27
4.2	Discrete-Time Extended Kalman Filter Algorithm . . . . .	28
4.3	Performance Measures for the EKF (Tuning) . . . . .	30
4.3.1	Zero-Mean Test on the Innovations $\underline{e}(t)$ . . . . .	31
4.3.2	Innovations Whiteness Test . . . . .	32
4.3.3	Weighted Sum Squared Residual (WSSR) . . . . .	33
4.3.4	“Reasonableness Tests ” on Covariances . . . . .	34
4.3.5	Tests When the True State Vector is Known . . . . .	34
<b>5</b>	<b>Simulation Experiment</b>	<b>35</b>
5.1	Choices for the Velocity and Acceleration Vectors $\underline{B}(t)$ and $\dot{\underline{B}}(t)$	35
5.2	Simulate the Accelerations $\underline{A}(t)$ . . . . .	37
5.3	Simulate the States $\underline{\omega}(t)$ Using the Gauss-Markov Model . . . . .	37
5.4	Estimate the States $\hat{\underline{\omega}}(k)$ Using the Extended Kalman Filter . . . . .	37
5.5	Examine the Performance of the EKF . . . . .	37
5.6	Mathematical/Numerical Performance Issues . . . . .	48
<b>6</b>	<b>Future Work</b>	<b>57</b>
6.1	Calibration and Error Compensation . . . . .	57
6.2	Perturbation Analysis . . . . .	57
6.3	Estimation Using Simulations of a Full Flight Profile . . . . .	58
6.4	Estimation Using Actual Flight Test Data . . . . .	58
<b>7</b>	<b>Conclusions</b>	<b>59</b>
	<b>Bibliography</b> . . . . .	<b>61</b>

# List of Figures

1.1	DAIMU sensor configuration. The six accelerometers are mounted at the centers of the faces of a cube. . . . .	6
1.2	DAIMU assembly . . . . .	7
1.3	Re-entry vehicle flight profile . . . . .	8
2.1	Coordinate system for the inertial and body frames . . . . .	13
2.2	Re-entry vehicle roll, pitch and yaw axes definition . . . . .	14
4.1	Block diagram depicting the re-entry vehicle kinematics and the Extended Kalman Filter . . . . .	29
5.1	(Exp.1) Simulated angular velocity $\underline{\omega}(k)$ and linear velocity $\underline{\nu}(k)$ (F1) . . . . .	38
5.2	(Exp.1) Simulated Accelerations $\dot{\underline{B}}(t)$ (F2) . . . . .	39
5.3	(Exp.1) Simulated Acceleration Measurements $\underline{A}(t)$ from the ODE Simulator (F3) . . . . .	40
5.4	(Exp.1) States $\underline{\omega}(t)$ Calculated by the Gauss-Markov Model (F22)	41
5.5	(Exp.1) States $\underline{\omega}(t)$ from the Gauss-Markov Model (Solid) and from the ODE Simulator (Dashed) (F26) . . . . .	42
5.6	(Exp.1) Measurements $\underline{y}(t)$ Calculated by the Gauss-Markov Model (F27) . . . . .	43
5.7	(Exp.1) Estimated states $\hat{\underline{\omega}}(k)$ from the EKF (solid lines) and simulated states $\underline{\omega}(k)$ from the Gauss- Markov model (dashed lines) (F38) . . . . .	44



5.8	(Exp.1) Estimated linear velocities $\hat{\underline{v}}(k)$ from quadrature on the estimated $\hat{\underline{p}}(k)$ obtained by using the $\hat{\underline{w}}(k)$ from the EKF in the ODE's (F49) . . . . .	45
5.9	(Exp.1) Estimated outputs $\hat{\underline{y}}(k)$ from the EKF (F41) . . . . .	46
5.10	(Exp.1) Simulated outputs $\underline{y}(k)$ from the Gauss- Markov model (solid lines) and estimated outputs $\hat{\underline{y}}(k)$ from the EKF (dashed lines) (F42) . . . . .	47
5.11	(Exp.1) Innovations $\underline{e}(k)$ and their “two-sigma bounds ”from the EKF (F43) . . . . .	49
5.12	(Exp.1) Whiteness test for the x-component of the innovations $e_x(k)$ (F44) . . . . .	50
5.13	(Exp.1) Whiteness test for the y-component of the innovations $e_y(k)$ (F45) . . . . .	51
5.14	(Exp.1) Whiteness test for the z-component of the innovations $e_z(k)$ (F46) . . . . .	52
5.15	(Exp.1) Weighted sum squared residuals (WSSR) (F43) . . . . .	53
5.16	(Exp.1) Centripetal acceleration $\underline{C}[\hat{\underline{w}}(k)]$ from the EKF (solid) and $\underline{C}[\underline{w}(k)]$ from the ODE simulator. Note that the accelerations are large. (F55) . . . . .	55
5.17	(Exp.1) Difference $\underline{A}(k) - \underline{C}[\underline{w}(k)]$ from the Gauss-Markov model (solid) and from the ODE simulator (dashed). This shows the numerical difficulty associated with the ODE's and the example problem. (F36) . . . . .	56

# Chapter 1

## Introduction

This work assesses the feasibility of using model-based nonlinear state estimation techniques to obtain the angular and linear velocities of a re-entry vehicle from six distributed accelerometer signals.

Conventional inertial navigation systems (INSs) use a gyroscope to measure angular velocity and three accelerometers to measure linear velocity. For a variety of technical reasons discussed below, it is sometimes desired to avoid using a gyroscope in an INS. Several researchers have shown the theoretical feasibility of avoiding gyroscopes by using signals from a specially designed array of six accelerometers to calculate *both* angular velocity and linear velocity for a rigid body in motion [1, 2, 3, 4, 5, 6]. The concept has yet to be fully demonstrated for a re-entry vehicle.

The Lawrence Livermore National Laboratory (LLNL) is conducting a program designed to use measurements from an accelerometer array in a re-entry vehicle [8, 10, 11, 9, 12, 13, 14]. The array is called the distributed accelerometer inertial measurement unit (DAIMU), and is designed based on the research reported in [1]. The array uses six accelerometers mounted on the faces of a cube as depicted in Figures (1.1) and (1.2) [8, 9].

### 1.1 Gyroscope vs. Accelerometer Array

*Gyroscopes* have the advantageous ability to measure angular velocity directly, without the need for special calculations. Gyroscope disadvantages include high cost, high power consumption, large volume, large weight, limited dynamic range in acceleration measurements, and long reaction time. *Accelerometer arrays* have the advantages of low cost, low power consumption, smaller volume, smaller weight, large dynamic range and short reaction time. In addition, the

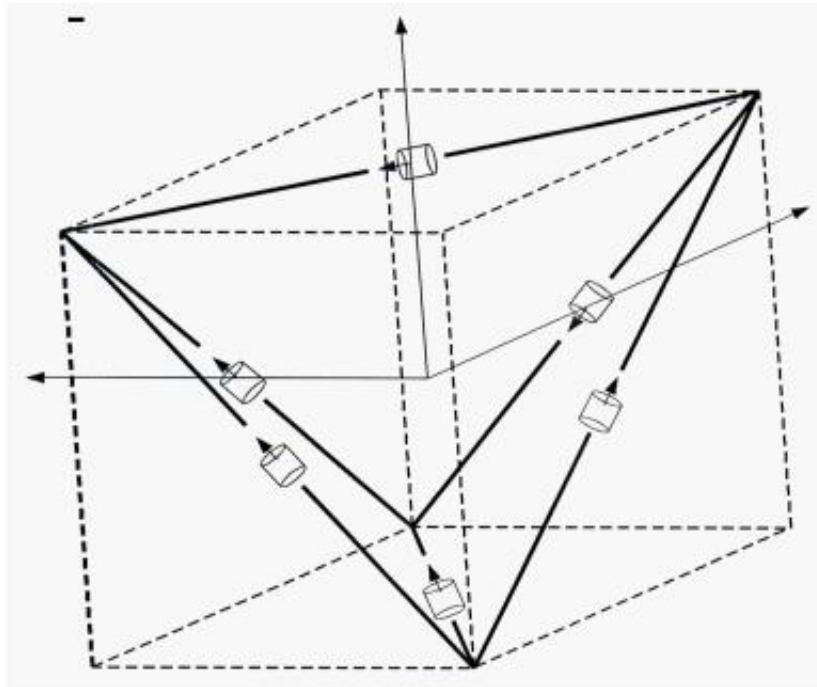


Figure 1.1: DAIMU sensor configuration. The six accelerometers are mounted at the centers of the faces of a cube.

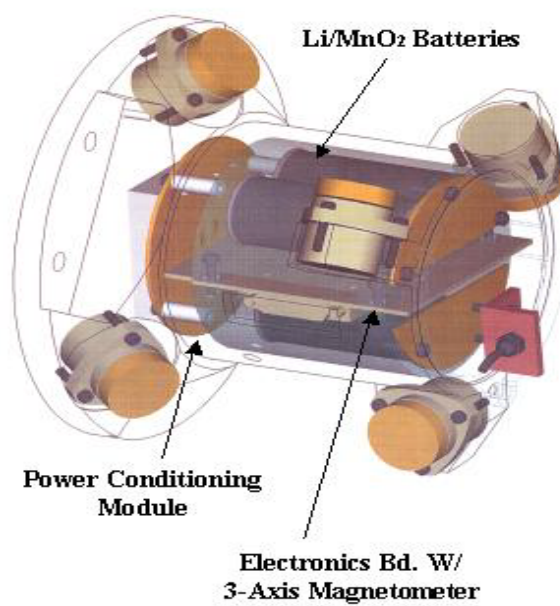
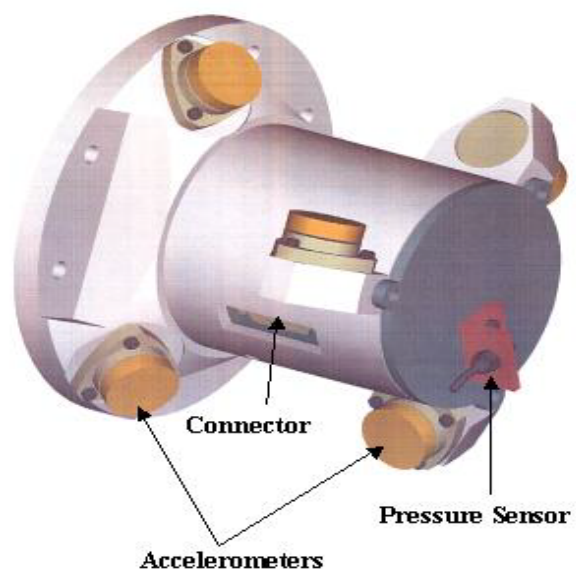


Figure 1.2: DAIMU assembly

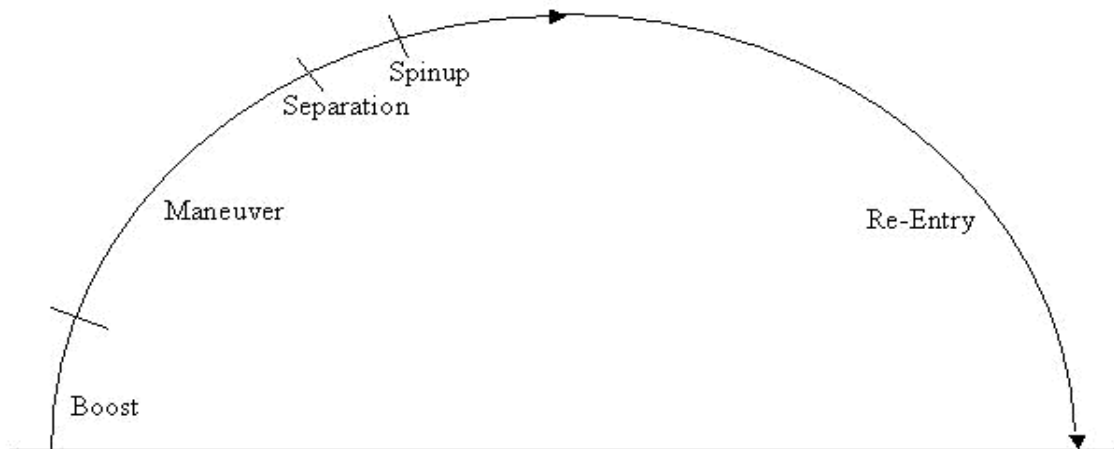


Figure 1.3: Re-entry vehicle flight profile

array can be distributed away from the center of action of the vehicle [8, 9]. The main disadvantages of accelerometer arrays are (1) Their velocity calculations contain inherent errors that grow nonlinearly with mission time and spin rate, due to fundamental mathematical/numerical issues associated with the system physics. For this reason, the application of an accelerometer-based velocity estimation scheme for a spinning re-entry vehicle presents *special challenges* not associated with other vehicles which do not spin. (2) For long mission times, it is generally necessary to use additional sensors (e.g. Global Positioning System (GPS) sensors) to bound navigation errors.

## 1.2 The DAIMU in a Re-Entry Vehicle

A flight profile for the re-entry vehicle is depicted simply in Figure (1.3). The vehicle system requirements and performance vary over this profile. At separation, the vehicle is gently released, so there is virtually no spin. Because of this, numerical issues in calculating velocities are not problematic. At spinup, the angular velocity of the vehicle reaches approximately 1.5 revolutions per second (rps). With this increased spin rate, numerical issues become problematic. At re-entry, the vehicle spins at approximately 13-17 rps or 22-25 rps depending on the re-entry angle, and the numerical issues are greatly exacerbated.

## 1.3 Theoretical Analysis

Given six signals from the DAIMU mounted in a re-entry vehicle, the goal of this study is to assess the feasibility of estimating the angular and linear velocities of the vehicle. Earlier work [12] showed that angular and linear velocity calculation for the DAIMU using classic ODE solvers is not practically feasible, due to mathematical and numerical difficulties. This report, describes an analysis that demonstrates the *theoretical feasibility* of using model-based nonlinear state estimation techniques to obtain the angular and linear velocities in this problem. Demonstrating *practical feasibility* of the estimation approach involves resolving numerical and calibration issues. For example, we must deal with uncertainty in the locations and sensing directions of the accelerometers (calibration to deal with mechanical tolerances). These issues will be the subjects of future reports.

The estimation of vehicle velocities is very difficult. The ordinary differential equations (ODEs) describing the kinematics of the vehicle are nonlinear, stiff and lead to exponentially growing spurious solutions containing drift/bias on very short time scales [12]. The solution depends nonlinearly on the initial angular velocity vector through a quadratic centripetal acceleration term. In addition, any noise on the measurements further disrupts solution attempts. Thus, classical numerical solution algorithms are not sufficient for obtaining use-

ful estimates of the angular and linear velocities over most useful time periods. Our proposed approach is to use state-space model-based nonlinear estimation techniques from the estimation theory, system theory and signal processing literature [17, 18, 19, 20, 21, 22, 23, 24, 26, 28, 29, 30, 31].

We first develop the six nonlinear ODEs for the vehicle kinematics [12, 15, 16] and show that they can be put in state-space form. To be useful for estimating the velocities, however, the state-space representation must be observable. A system is said to be observable if measurements of the outputs contain sufficient information to enable us to identify the internal system states  $\mathbf{x}(t)$  [26, 28]. We analyze the properties of the state-space system, and show that six accelerometer measurements alone are *not* sufficient to ensure observability of the system. We show, however, that observability could be achieved, if measurements of the angular velocities were available (e.g. Global Positioning System (GPS) signals). Unfortunately, programmatic and technical constraints preclude the use of GPS with our vehicle [10].

Next, we show that the system can be made observable by augmenting the acceleration measurements with angular velocity measurements from an existing three 3-axis magnetometer currently mounted in the DAIMU assembly. This magnetometer was originally placed in the assembly for other reasons, but it can now serve a dual purpose [10]. We show that we can decouple the six ODEs into three ODEs for the angular velocities and three ODEs for the linear velocities. We then show that once the angular velocities are available from an estimator, the three ODEs for the linear velocity become algebraic; and the linear velocities can be calculated directly by numerical integration. This property allows us to make the system observable by using only magnetometer measurements of angular velocity. The need for linear velocity measurements is obviated, at least in theory. GPS is not necessary. This observation greatly simplifies the mathematics and minimizes the number of sensors required in the vehicle. Thus, the estimation method is *theoretically feasible*.

## 1.4 Estimator Development and Testing

In the following sections, we present the development of a discrete-time Gauss-Markov model and an Extended Kalman Filter (EKF) for estimating the velocities. Using a simple simulation example, we then show that the estimator adds value over the numerical ODE solver in the presence of measurement noise. The EKF is able to deal effectively with the noise and provide useful angular velocity estimates. The linear velocity estimates for this simulation show numerical difficulties associated with the nonlinear ODE's and the quadrature operation. The final section describes *practical numerical and calibration issues* that must be addressed in future work.

## Chapter 2

# Kinematic Equations for the Re-Entry Vehicle

The kinematic formulation is for a three-dimensional inertial reference frame [15, 16] denoted  $O_I$  and depicted in Figure (2.1). The body frame, labeled  $O_B$  represents the frame for the re-entry vehicle. The figure depicts the case in which there is a single accelerometer located at point  $\mathbf{P}$ . Figure (2.2) defines the roll, pitch and yaw axes.

We use the following notation. Vectors are generally underlined (e.g.  $\underline{\omega}(t)$ ). Unit vectors and sensing direction vectors, however, are deemed special and have a “hat ” above them rather than a line under them (e.g.  $\hat{\theta}$ ). Tensors/matrices generally use capital characters and have a tilde above them (e.g.  $\tilde{T}$ ). In the section on Kalman Filtering, the hat and tilde take on additional meanings.

The 3X1 body frame vector  $\underline{r}_p$  for the position of accelerometer p ( $p = 1, 2, \dots, 6$ ) is given by:

$$\underline{r}_p \triangleq r_1 \hat{e}_1 + r_2 \hat{e}_2 + r_3 \hat{e}_3 \quad (2.1)$$

$$\triangleq [r_1 \ r_2 \ r_3]^T \quad (3X1) \text{ Accelerometer position} \quad (2.2)$$

where  $\hat{e}_1, \hat{e}_2, \hat{e}_3$  are unit vectors along the orthogonal axes of the body frame  $O_B$ ,  $\hat{e}_1 = [1 \ 0 \ 0]^T$ ,  $\hat{e}_2 = [0 \ 1 \ 0]^T$ , and  $\hat{e}_3 = [0 \ 0 \ 1]^T$ . The 3X1 time-varying angular velocity vector for the body is given by:

$$\underline{\omega}(t) \triangleq \omega_1(t) \hat{e}_1 + \omega_2(t) \hat{e}_2 + \omega_3(t) \hat{e}_3 \quad (2.3)$$

$$\triangleq [\omega_1(t) \ \omega_2(t) \ \omega_3(t)]^T \quad (3X1) \text{ Angular velocities} \quad (2.4)$$



The 3X1 time-varying linear velocity vector for the body is given by:

$$\underline{\nu}(t) \triangleq \nu_1(t)\hat{e}_1 + \nu_2(t)\hat{e}_2 + \nu_3(t)\hat{e}_3 \quad (2.5)$$

$$\triangleq [\nu_1(t) \ \nu_2(t) \ \nu_3(t)]^T \quad (3X1) \text{ Linear velocities} \quad (2.6)$$

For a single accelerometer, the 3X1 accelerometer sensing direction vector  $\hat{\theta}$  is given by:

$$\hat{\theta} \triangleq \theta_1\hat{e}_1 + \theta_2\hat{e}_2 + \theta_3\hat{e}_3 \quad (2.7)$$

where  $\hat{\theta}$  is determined by the designed location of the accelerometer in the DAIMU fixture. The symbols “ $\cdot$ ” and “ $\times$ ” denote the vector inner product and vector cross product, respectively. Given the definitions above, and letting  $t$  denote the continuous time variable, we can write the acceleration of point  $\mathbf{P}$  as follows [15, 16]:

$$\ddot{\underline{R}}_p(t) = \ddot{\underline{R}}_b(t) + \ddot{\underline{r}}_p(t) + \dot{\underline{\omega}}(t) \times \underline{r}_p(t) + 2\underline{\omega}(t) \times \dot{\underline{r}}_p(t) + \underline{\omega}(t)[\underline{\omega}(t) \times \underline{r}_p(t)] \quad (2.8)$$

For a single accelerometer mounted at point  $\mathbf{P}$  on the body, we have  $\dot{\underline{r}}_p(t) = 0$ , because the accelerometer is not moving relative to the body frame  $O_B$ . In this case, we can write:

$$\ddot{\underline{R}}_p(t) = \ddot{\underline{R}}_b(t) + \dot{\underline{\omega}}(t) \times \underline{r}_p(t) + \underline{\omega}(t)[\underline{\omega}(t) \times \underline{r}_p(t)] \quad (2.9)$$

Given the accelerometer sensing direction vector  $\hat{\theta}$  and defining the acceleration  $\ddot{\underline{R}}_p(t) \cdot \hat{\theta} \triangleq \underline{A}(t)$ , we can write

$$\underline{A}(t) = \ddot{\underline{R}}_p(t) \cdot \hat{\theta} \quad (2.10)$$

$$= \ddot{\underline{R}}_b(t) \cdot \hat{\theta} + [\dot{\underline{\omega}}(t) \times \underline{r}_p(t)] \cdot \hat{\theta} + \{\underline{\omega}(t)[\underline{\omega}(t) \times \underline{r}_p(t)]\} \cdot \hat{\theta} \quad (2.11)$$

To simplify the notation, we can drop the b and p subscripts so  $\ddot{\underline{R}}(t) \triangleq \ddot{\underline{R}}_p(t)$  and  $\underline{r}(t) \triangleq \underline{r}_p(t)$ . We then define the linear velocity vector  $\dot{\underline{r}}(t) \triangleq \dot{\underline{R}}(t)$  and write the kinematic equation as

$$\underline{A}(t) = \dot{\underline{r}}(t) \cdot \hat{\theta} + [\dot{\underline{\omega}}(t) \times \underline{r}(t)] \cdot \hat{\theta} + \{\underline{\omega}(t)[\underline{\omega}(t) \times \underline{r}(t)]\} \cdot \hat{\theta} \quad (6X1) \quad (2.12)$$

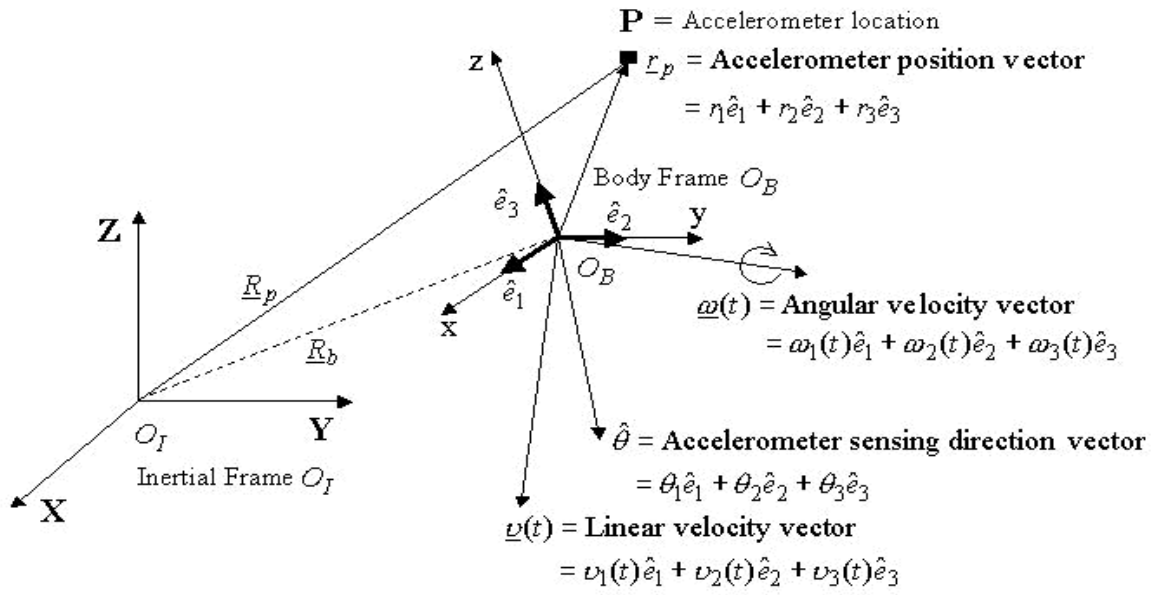


Figure 2.1: Coordinate system for the inertial and body frames

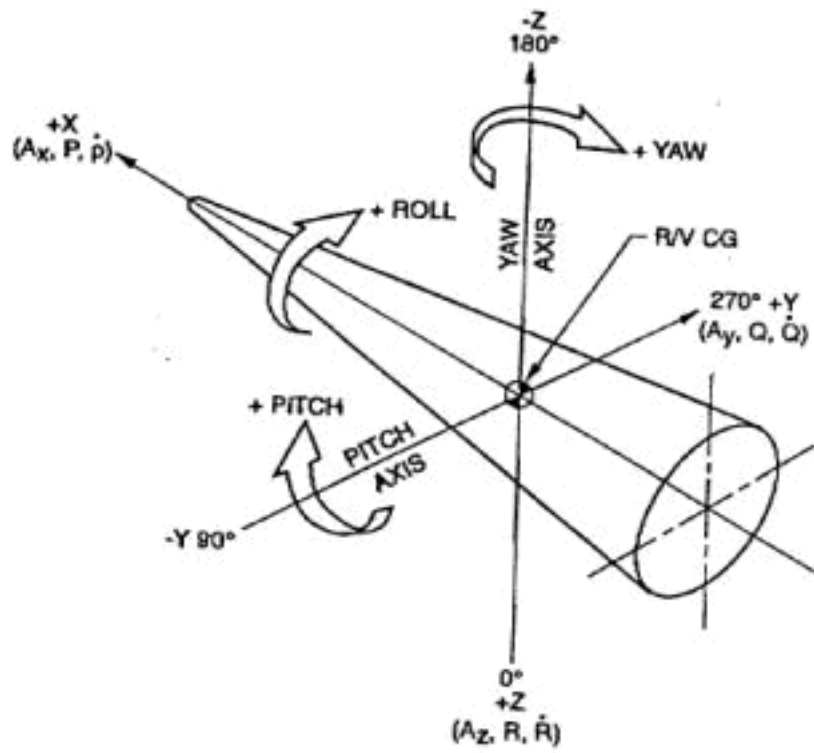


Figure 2.2: Re-entry vehicle roll, pitch and yaw axes definition

The equation is linear in the first term (the linear acceleration), linear in the second term (the angular acceleration) and quadratic (nonlinear) in the third term (the centripetal acceleration). As we shall see later, this nonlinearity has a great impact upon our ability to solve the ODE's and build an estimator for the velocities.

Given an array of six accelerometers, it has been shown [1] that it is theoretically possible to determine  $\underline{A}(t)$  and  $\underline{\dot{\omega}}(t)$  from six measured accelerations  $A_n(t)$ ,  $n = 1, 2, \dots, 6$ . Then, given initial conditions  $\underline{\omega}(0)$ , we can integrate  $\underline{\dot{\omega}}(t)$ , to obtain  $\underline{\omega}(t)$ .

Dr. David H. Chambers of the Lawrence Livermore National Laboratory (LLNL) has performed theoretical and computational analyses of this set of nonlinear ODE's [12]. The ODE's are numerically stiff. Numerical solutions have exponentially growing spurious solutions with very short time scales. The spurious solutions can be eliminated by using an implicit or semi-implicit numerical integration scheme. The semi-implicit numerical integration scheme used in this analysis is second order accurate in time. The solution, however, depends nonlinearly on the initial angular velocity vector  $\underline{\omega}(0)$  through the quadratic centripetal acceleration term. The solutions are sensitive to stochastic measurement noise, and problems arise with very low-frequency drift or bias in the acceleration measurements. Given these difficulties, it has been suggested that a modern estimation approach using the kinematic model and allowing for stochastic measurement noise may offer an advantage in estimating the velocities. This is the motivation for the work in this paper. We next develop the kinematic equations in state- space form in preparation for the development of an estimation algorithm.

## Chapter 3

# Development of the Kinematic System Equations in State Space Form

Let us now express the kinematic equations above in state-space form, because it facilitates our use of modern estimation techniques [17, 20, 26].

In order to describe all six accelerometer locations, we define the 3X1 vector  $\hat{\theta}_m$  to be the sensing direction vector for accelerometer  $m$ ,  $m = 1, 2, \dots, 6$ . It represents the direction in which the accelerometer is mounted relative to the body frame  $O_B$ .

$$\hat{\theta}_m \triangleq \theta_{m1}\hat{e}_1 + \theta_{m2}\hat{e}_2 + \theta_{m3}\hat{e}_3 \quad (3.1)$$

$$\triangleq [\theta_{m1} \ \theta_{m2} \ \theta_{m3}]^T \quad m = 1, 2, \dots, 6 \quad (3X1) \text{ Sensing direction} \quad (3.2)$$

We define the following signal vectors used to describe the system kinematics:

$$\underline{A}(t) \triangleq [A_1(t) \ A_2(t) \ \dots \ A_6(t)]^T \quad (6X1) \text{ Measured Accelerations} \quad (3.3)$$

$$\underline{B}(t) \triangleq [\omega_1(t) \ \omega_2(t) \ \omega_3(t) \ \nu_1(t) \ \nu_2(t) \ \nu_3(t)]^T \quad (3.4)$$

$$\triangleq [\underline{\omega}^T(t) \ \underline{\nu}^T(t)]^T \quad (6X1) \text{ Velocities} \quad (3.5)$$

### 3.1 The Vector ODE Relating $\underline{A}(t)$ and $\dot{\underline{B}}(t)$

We can write a vector ordinary differential equation (ODE) describing the kinematics of the system as follows:

$$\underline{A}(t) = \tilde{T} \dot{\underline{B}}(t) + \underline{C}[\underline{B}(t)] \quad (3.6)$$

where  $\tilde{T}$  is a 6X6 constant matrix designed to be nonsingular and determined by the spatial configuration of the accelerometer array in the DAIMU [1, 9, 12]. It is important to note that  $\underline{C}[\underline{B}(t)]$  is a *quadratic* function of the angular velocities  $\underline{\omega}(t)$ , but it is *NOT* a function of the linear velocities  $\underline{\nu}(t)$ . We show the details of  $\tilde{T}$  and  $\underline{C}[\underline{B}(t)]$  in the following sections.

Because we shall be working in state-space, it is convenient to rearrange the vector ODE as follows:

$$\dot{\underline{B}}(t) = \tilde{T}^{-1} \{ \underline{A}(t) - \underline{C}[\underline{B}(t)] \} \quad (6X1) \quad (3.7)$$

or

$$\dot{\underline{B}}(t) = -\tilde{T}^{-1} \underline{C}[\underline{B}(t)] + \tilde{T}^{-1} \underline{A}(t) \quad (6X1) \quad (3.8)$$

#### 3.1.1 The Configuration Matrix $\tilde{T}$

With some manipulation of the basic kinematic equations and by using the definitions of the coordinate system(s), we define the important matrices  $\tilde{T}$  and  $\underline{C}[\underline{B}(t)]$  as follows:

$$\tilde{T} = [\tilde{T}^{(1)} \ \tilde{T}^{(2)}] \quad (6X6) \text{ Configuration Matrix} \quad (3.9)$$

where  $\tilde{T}^{(1)}$  and  $\tilde{T}^{(2)}$  are 6X3 matrices with elements defined as follows:

$$T^{(1)}_{mn} = (\underline{r}_m \times \hat{\theta}_m) \cdot \hat{e}_n \quad m = 1, 2, \dots, 6 \quad n = 1, 2, 3 \quad (3.10)$$

$$T^{(2)}_{mn} = \hat{\theta}_m \cdot \hat{e}_n \quad m = 1, 2, \dots, 6 \quad n = 1, 2, 3 \quad (3.11)$$

This partitioning of the configuration matrix is convenient because we can think of  $\tilde{T}^{(1)}$  as the 6X3 part of the matrix related to the angular acceleration  $\underline{\dot{\omega}}$  and  $\tilde{T}^{(2)}$  as the 6X3 part of the matrix related to the linear acceleration  $\underline{\dot{v}}$ .

For increased understanding and convenience, we can partition  $\tilde{T}^{(1)}$  into three 6X1 column vectors  $\underline{T}^{(1)}_{mn}$ ,  $n = 1, 2, 3$  as follows:

$$\tilde{T}^{(1)} = [\underline{T}^{(1)}_{m1} \quad \underline{T}^{(1)}_{m2} \quad \underline{T}^{(1)}_{m3}] \quad (3.12)$$

$$= \begin{pmatrix} T^{(1)}_{11} & T^{(1)}_{12} & T^{(1)}_{13} \\ T^{(1)}_{21} & T^{(1)}_{22} & T^{(1)}_{23} \\ \vdots & \vdots & \vdots \\ T^{(1)}_{61} & T^{(1)}_{62} & T^{(1)}_{63} \end{pmatrix} \quad (6X3) \quad (3.13)$$

where the elements of the three column vectors can be written:

$$T^{(1)}_{m1} = r_{m2}\theta_{m3} - r_{m3}\theta_{m2} \quad m = 1, 2, \dots, 6 \quad (3.14)$$

$$T^{(1)}_{m2} = r_{m3}\theta_{m1} - r_{m1}\theta_{m3} \quad m = 1, 2, \dots, 6 \quad (3.15)$$

$$T^{(1)}_{m3} = r_{m1}\theta_{m2} - r_{m2}\theta_{m1} \quad m = 1, 2, \dots, 6 \quad (3.16)$$

As before, for increased understanding and convenience, we can partition  $\tilde{T}^{(2)}$  into six 3X1 row vectors  $\hat{\theta}_m^T$ ,  $m = 1, 2, \dots, 6$  as follows:

$$\tilde{T}^{(2)} = \begin{pmatrix} \hat{\theta}_1^T \\ \hat{\theta}_2^T \\ \vdots \\ \hat{\theta}_6^T \end{pmatrix} \quad (6X3) \quad (3.17)$$

where the six 1X3 row vectors can be written:

$$\hat{\theta}_m^T = [\theta_{m1} \quad \theta_{m2} \quad \theta_{m3}] \quad m = 1, 2, \dots, 6 \quad (1X3) \quad (3.18)$$

### 3.1.2 The Centripetal Acceleration Vector $\underline{C}[\underline{B}(t)]$

The scalar centripetal acceleration term of the kinematic equation (2.12) is given as follows:

$$\{\underline{\omega}(t) \times [\underline{\omega}(t) \times \underline{r}_m]\} \cdot \hat{\theta}_m = \{[\underline{\omega}(t) \cdot \underline{r}_m]\underline{\omega}(t) - [\underline{\omega}(t) \cdot \underline{\omega}(t)]\underline{r}_m\} \cdot \hat{\theta}_m \quad (3.19)$$

$$= [\underline{\omega}(t) \cdot \underline{r}_m][\hat{\theta}_m \cdot \underline{\omega}(t)] - [\underline{\omega}(t) \cdot \underline{\omega}(t)][\hat{\theta}_m \cdot \underline{r}_m] \quad (3.20)$$

$$= C_m[\underline{\omega}(t)] \quad (3.21)$$

for  $m = 1, 2, \dots, 6$ . Recognizing that the centripetal acceleration vector  $\underline{C}[\underline{B}(t)]$  is a function only of the angular velocities  $\underline{\omega}(t)$  and not the linear velocities  $\underline{v}(t)$ , we can substitute  $\underline{C}[\underline{\omega}(t)]$  for  $\underline{C}[\underline{B}(t)]$ .

$$\underline{C}[\underline{B}(t)] = \underline{C}[\underline{\omega}(t)] = \begin{pmatrix} C_1[\underline{\omega}(t)] \\ C_2[\underline{\omega}(t)] \\ \vdots \\ C_6[\underline{\omega}(t)] \end{pmatrix} \quad (6X1) \quad (3.22)$$

Further, using another notation for the vector inner product, we can write the six elements of the centripetal acceleration vector as follows for  $m = 1, 2, \dots, 6$ :

$$C_m[\underline{\omega}(t)] = [\underline{\omega}^T(t)\underline{r}_m][\underline{\omega}^T(t)\hat{\theta}_m] - [\underline{\omega}^T(t)\underline{\omega}(t)][\underline{r}_m^T\hat{\theta}_m] \quad (\text{scalar}) \quad (3.23)$$

Note that when programming this equation, there is a computational advantage to substituting a constant  $\omega^2$  for the quantity  $\underline{\omega}^T(t)\underline{\omega}(t)$ . In addition, this equation for  $C_m$  can be written in a form that is sometimes more convenient and useful, as we show next. In general, for 3X1 vectors  $\underline{x}$  and  $\underline{y}$ , it can be shown that their cross product can be written as a matrix multiplication using a skew-symmetric matrix  $\Omega$  as follows:



$$\underline{x} \times \underline{y} = \begin{pmatrix} 0 & y_3 & -y_2 \\ -y_3 & 0 & y_1 \\ y_2 & -y_1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \tilde{\Omega} \underline{x} \quad (3.24)$$

$$= \begin{vmatrix} \hat{e}_1 & \hat{e}_2 & \hat{e}_3 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix} \quad (3X1) \quad (3.25)$$

Note that because  $\tilde{\Omega}$  is skew-symmetric [35], it has the property that  $\tilde{\Omega}^T = -\tilde{\Omega}$ . For our problem, we define  $\tilde{\Omega}$  as follows:

$$\tilde{\Omega}[\underline{\omega}(t)] \triangleq \begin{pmatrix} 0 & \omega_3(t) & -\omega_2(t) \\ -\omega_3(t) & 0 & \omega_1(t) \\ \omega_2(t) & -\omega_1(t) & 0 \end{pmatrix} \quad (3X3) \quad (3.26)$$

Using these identities, we can show that:

$$C_m[\underline{\omega}(t)] = \{\underline{\omega}(t) \times [\underline{\omega}(t) \times \underline{r}_m]\} \cdot \hat{\theta}_m \quad (3.27)$$

$$= [\underline{\omega}^T(t) \underline{r}_m] [\underline{\omega}^T(t) \hat{\theta}_m] - [\underline{\omega}^T(t) \underline{\omega}(t)] [\underline{r}_m^T \hat{\theta}_m] \quad (3.28)$$

$$= \{\tilde{\Omega}^2[\underline{\omega}(t)] \underline{r}_m\}^T \hat{\theta}_m \quad (3.29)$$

$$= \underline{r}_m^T \tilde{\Omega}^2[\underline{\omega}(t)] \hat{\theta}_m \quad (3X1) \quad (3.30)$$

### 3.1.3 Decoupling the Continuous-Time Kinematic Equations

Noting that the centripetal acceleration term  $\underline{C}[B(t)]$  is really a function only of the angular velocity, we can write the basic kinematic equation as follows:

$$\dot{\underline{B}}(t) = -\tilde{T}^{-1} \underline{C}[\underline{\omega}(t)] + \tilde{T}^{-1} \underline{A}(t) \quad (3.31)$$

Exploiting the above-mentioned property of  $\underline{C}[B(t)]$ , we can decouple the set of six ODE's into a set of three equations in the linear velocity and three equations in the angular velocity. We do this by defining the 3X3 matrix  $\tilde{\Gamma}$  to be the inverse of the configuration matrix,  $\tilde{T}$ , and partitioning it into a 3X6 upper part  $\tilde{\Gamma}_U$  and a 3X6 lower part  $\tilde{\Gamma}_D$  as follows:

$$\tilde{\Gamma} \triangleq \tilde{T}^{-1} = \begin{bmatrix} \tilde{\Gamma}_U \\ \tilde{\Gamma}_D \end{bmatrix} \quad (6X6) \quad (3.32)$$

Using this definition, the equations become

$$\underline{\dot{B}}(t) = -\tilde{\Gamma}\underline{C}[\underline{\omega}(t)] + \tilde{\Gamma}\underline{A}(t) \quad (3.33)$$

We can see that these equations are decoupled by expanding it as follows:

$$\begin{bmatrix} \underline{\dot{\omega}}(t) \\ \underline{\dot{L}}(t) \end{bmatrix} = -\begin{bmatrix} \tilde{\Gamma}_U \\ \tilde{\Gamma}_D \end{bmatrix} \underline{C}[\underline{\omega}(t)] + \begin{bmatrix} \tilde{\Gamma}_U \\ \tilde{\Gamma}_D \end{bmatrix} \underline{A}(t) \quad (6X1) \quad (3.34)$$

This decoupling operation has a great advantage for us in this application, because it allows us to reduce the dimension of the state space from 6 to 3. By inspection, we see that we can build an optimal nonlinear estimator using the 3X1 equation in the angular velocity to estimate  $\underline{\hat{\omega}}(t)$ . Given these estimates, we can integrate the 3X1 equation in the linear velocity to obtain  $\underline{\hat{L}}(t)$ . We do not need to build a 6-dimensional estimator.

## 3.2 Formulating the Kinematic Equations as an Observable Continuous-Time Gauss-Markov System Model

Even though we have created a state-space representation for the kinematic ODE's derived earlier, system theory tells us that development of a state estimator requires that the state-space system must be *observable* [26, 28]. Simply stated, a system is observable if measurements of its outputs are sufficient to enable us to identify completely the states. The system we have developed above does not meet this criterion. The only measurements we have are those from the accelerometers, and these do not help us evaluate the state vector  $\underline{\hat{B}}(t)$ . The system is not observable.

Fortunately, we are able to make the system observable by using two important pieces of information: (1) The system can be de-coupled as noted above, so we do not need measurements of the linear velocities. We need only measurements of the angular velocities, for example from a Global Positioning System (GPS). (2) Unfortunately, the constraints of the program dictate that we cannot use GPS, so we must find an alternative. Fortunately, we have on board the re-entry vehicle a three-axis magnetometer, from which we can obtain measurements of the angular velocity vector  $\underline{\omega}(t)$ .

We can now specify a continuous-time state-space Gauss-Markov model for the system as follows:

### 3.2.1 System Plant

$$\dot{\underline{\omega}}(t) = -\tilde{\Gamma}_U \underline{C}[\underline{\omega}(t)] + \tilde{\Gamma}_U \underline{A}(t) + \underline{W}(t) \quad (3.35)$$

where  $\underline{W}(t)$  is a 3X1 vector of white Gaussian noise (WGN) distributed with mean zero and covariance  $\tilde{R}_W$ ,  $\underline{W}(t) \sim N[0, \tilde{R}_W]$ .

### 3.2.2 System Measurements

The system output measurement equation is obtained simply from the magnetometer measurements of the angular velocities  $\underline{\omega}(t)$ , with additive white Gaussian measurement noise assumed.

$$\underline{y}(t) = \underline{\omega}(t) + \underline{V}(t) \quad (3X1) \quad (3.36)$$

where  $\underline{V}(t)$  is a 3X1 vector of white Gaussian noise distributed with mean zero and covariance  $\tilde{R}_V$ ,  $\underline{V}(t) \sim N[0, \tilde{R}_V]$ .

Note that it is possible for the magnetometer measurements to be null momentarily when the magnetometer is aligned perfectly with the magnetic field lines of the Earth. However, the probability of the magnetometer being aligned perfectly with the Earth's magnetic field for any extended period of time is small. In addition, if the data were to include momentary "drop-outs" because of this, we could post-process the data to interpolate samples, etc. Our engineering judgment is that this pathological situation is unlikely and not a severe problem if it does occur.

### 3.2.3 Estimation of the Linear Velocities

The Gauss-Markov model given above is useful for forming the basis of an Extended Kalman Filter (EKF) [17, 19, 24, 20, 29, 31] to estimate the three angular velocities  $\underline{\omega}(k)$ . We can exploit the fact that the ODE's are decoupled to estimate the linear velocities  $\underline{v}(t)$  directly. Once we have the estimated angular velocities  $\hat{\underline{\omega}}(t)$  from the EKF, the 3 ODE's for the linear velocities become algebraic, so we can calculate  $\hat{\underline{v}}(t)$  directly using the continuous-time form of the ODE's as follows:

$$\hat{\underline{v}}(t) = -\tilde{\Gamma}_D \underline{C}[\hat{\underline{\omega}}(t)] + \tilde{\Gamma}_D \underline{A}(t) \quad (3X1) \quad (3.37)$$

We can then integrate  $\hat{\underline{v}}(t)$  using a numerical quadrature algorithm to estimate  $\hat{\underline{v}}(t)$ . With estimates of both the angular and linear velocities, we now have an estimate of the full state vector  $\underline{B}(t)$ .

### 3.3 Discrete-Time Gauss-Markov Model

For implementation on a digital computer, we need a discrete-time equivalent of the continuous-time Gauss-Markov model developed in the last section. We have at least two options for accomplishing this: (1) We can use the continuous-discrete formulation of the EKF [17, 20]. For this algorithm, the ODE's are integrated using an ODE solver and the state estimates are computed using a discrete-time filter formulation. (2) We can discretize the continuous-time Gauss-Markov model and use the discrete-discrete EKF formulation, in which both the Gauss-Markov model and the filter are implemented in discrete-time. We choose to use the latter option, because the discretization of the model can be accomplished reasonably, as follows.

#### 3.3.1 Discretization Using a 2nd-Order Semi-Implicit Method

A discrete-time Gauss-Markov model [17, 20] can be created by de-coupling the ODE's and discretizing them using a second-order semi-implicit integration scheme [37, 36]. First, define the quantity  $\underline{F}[\underline{\omega}, t]$  as follows:

$$\dot{\underline{\omega}}(t) = -\tilde{\Gamma}_U \underline{C}[\underline{\omega}(t)] + \tilde{\Gamma}_U \underline{A}(t) + \underline{W}(t) \equiv \underline{F}[\underline{\omega}, t] \quad (3.38)$$

Next, apply the second-order semi-implicit integration method [37] to this equation, denoting the iteration index by  $n = 1, 2, \dots$

$$\underline{\omega}^{n+1} = \underline{\omega}^n + \frac{1}{2} \Delta t [\underline{F}(\underline{\omega}^n, t_{n+1}) + \underline{F}(\underline{\omega}^n, t_n) + \tilde{G}(\underline{\omega}^n, t_n)(\underline{\omega}^{n+1} - \underline{\omega}^n)] \quad (3.39)$$

The 3X3 Jacobian matrix  $\tilde{G}(\underline{\omega}^n)$  is defined by the following:

$$\tilde{G}[\underline{\omega}^n] \triangleq -\tilde{\Gamma}_U \frac{\partial \underline{C}[\underline{\omega}^n]}{\partial \underline{\omega}} \bigg|_{\underline{\omega}=\underline{\omega}^n} \quad (3X3) \text{ Jacobian Matrix} \quad (3.40)$$

Expanding gives:

$$\underline{\omega}^{n+1} = \underline{\omega}^n + \Delta t [\tilde{I} - \frac{1}{2} \Delta t \tilde{G}(\underline{\omega}^n)]^{-1} \tilde{\Gamma}_U [\frac{1}{2} (\underline{A}^{n+1} + \underline{A}^n) - \underline{C}(\underline{\omega}^n)] \quad (3.41)$$

We can simplify this equation by making a reasonable and commonly-used approximation; We can replace the term  $\frac{1}{2}(\underline{A}^{n+1} + \underline{A}^n)$  with  $\underline{A}^n$ . Additionally, we define the 3X3 matrix  $\tilde{\Psi}[\underline{\omega}(k)]$  as follows:

$$\tilde{\Psi}^n \triangleq \Delta t [\tilde{I} - \frac{1}{2} \Delta t \tilde{G}(\underline{\omega}^n)]^{-1} \quad (3X3) \quad (3.42)$$

where  $\Delta t$  is the data sampling period in seconds. Expanding yields

$$\underline{\omega}^{n+1} = \underline{\omega}^n - \tilde{\Psi}^n \tilde{\Gamma}_U \underline{C}(\underline{\omega}^n) + \tilde{\Psi}^n \tilde{\Gamma}_U \underline{A}^n \quad (3X1) \quad (3.43)$$

### 3.3.2 Discretized System Plant

To formulate the discretized Gauss-Markov model using the results from the last section, let the discrete time index  $k = 1, 2, \dots$  equal the iteration index  $n$ . We define the 3X3 matrix  $\tilde{\Psi}[\underline{\omega}(k)]$  as follows:

$$\tilde{\Psi}[\underline{\omega}(k)] \triangleq \Delta t \{ \tilde{I} - \frac{1}{2} \tilde{G}[\underline{\omega}(k)] \}^{-1} \quad (3X3) \quad (3.44)$$

where the 3X3 Jacobian matrix  $\tilde{G}(k)$  is defined by the following:

$$\tilde{G}[\underline{\omega}(k)] \triangleq -\tilde{\Gamma}_U \left. \frac{\partial \underline{C}[\underline{\omega}(k)]}{\partial \underline{\omega}(k)} \right|_{\underline{\omega}=\underline{\omega}(k)} \quad (3X3) \text{ Jacobian Matrix} \quad (3.45)$$

The 6X3 partial derivative in the Jacobian can be written as follows:

$$\frac{\partial \underline{C}[\underline{\omega}(k)]}{\partial \underline{\omega}^T(k)} = \begin{pmatrix} \frac{\partial C_1[\underline{\omega}(k)]}{\partial \omega_1(k)} & \frac{\partial C_1[\underline{\omega}(k)]}{\partial \omega_2(k)} & \frac{\partial C_1[\underline{\omega}(k)]}{\partial \omega_3(k)} \\ \frac{\partial C_2[\underline{\omega}(k)]}{\partial \omega_1(k)} & \frac{\partial C_2[\underline{\omega}(k)]}{\partial \omega_2(k)} & \frac{\partial C_2[\underline{\omega}(k)]}{\partial \omega_3(k)} \\ \vdots & \vdots & \vdots \\ \frac{\partial C_6[\underline{\omega}(k)]}{\partial \omega_1(k)} & \frac{\partial C_6[\underline{\omega}(k)]}{\partial \omega_2(k)} & \frac{\partial C_6[\underline{\omega}(k)]}{\partial \omega_3(k)} \end{pmatrix} \quad (6X3) \quad (3.46)$$

Using the results derived earlier, we see that for the re-entry vehicle, each 1X3 row of this Jacobian matrix can be written as follows ( $m = 1, 2, \dots, 6$ ):

$$\frac{\partial C_m[\underline{\omega}(k)]}{\partial \underline{\omega}^T(k)} = [\underline{\omega}^T(k) \underline{r}_m] \hat{\theta}_m^T + [\underline{\omega}^T(k) \hat{\theta}_m] \underline{r}_m^T - 2 \underline{\omega}^T(k) [\underline{r}_m^T \hat{\theta}_m] \quad (1X3) \quad (3.47)$$

Given these definitions, the discrete-time Gauss-Markov model for the states can be written

$$\underline{\omega}(k+1) = \underline{\omega}(k) - \tilde{\Psi}[\underline{\omega}(k)] \tilde{\Gamma}_U \underline{C}[\underline{\omega}(k)] + \tilde{\Psi}[\underline{\omega}(k)] \tilde{\Gamma}_U \underline{A}(k) + \underline{W}(k) \quad (3X1) \quad (3.48)$$

where  $\underline{W}(k)$  is a 3X1 vector of white Gaussian noise (WGN) distributed with mean zero and covariance  $\tilde{R}_W$ ,  $\underline{W}(t) \sim N[0, \tilde{R}_W]$ .

### 3.3.3 Discrete-Time System Output Measurements:

The system output measurement equation is obtained simply from the magnetometer measurements of the angular velocities  $\underline{\omega}(k)$ , with additive white Gaussian noise assumed.

$$\underline{y}(k) = \underline{\omega}(k) + \underline{V}(k) \quad (3X1) \quad (3.49)$$

where  $\underline{V}(k)$  is a 3X1 vector of white Gaussian noise distributed with mean zero and covariance  $\tilde{R}_V$ ,  $\underline{V}(k) \sim N[0, \tilde{R}_V]$ .

### 3.3.4 Estimation of the Linear Velocities $\underline{v}(k)$

The Gauss-Markov model given above is useful for forming the basis of an Extended Kalman Filter (EKF) to estimate the three angular velocities  $\underline{\omega}(k)$ . We can use the fact that the ODE's are decoupled to our advantage in estimating the linear velocities  $\underline{v}(k)$ . Given the estimated angular velocities  $\hat{\underline{\omega}}(k)$  from the EKF, we can calculate  $\hat{\underline{v}}(k)$  directly using the continuous-time form of the ODE's as follows:

$$\hat{\underline{v}}(k) = -\tilde{\Gamma}_D \underline{C}[\hat{\underline{\omega}}(k)] + \tilde{\Gamma}_D \underline{A}(k) \quad (3X1) \quad (3.50)$$

We can then integrate (sum) the  $\hat{\underline{v}}(k)$  to estimate  $\hat{\underline{v}}(k)$ . In practice, we use a recursive trapezoidal rule algorithm [37] as follows:

$$\hat{v}(k) = \hat{v}(k-1) + \frac{1}{2}\Delta t[\hat{v}(k-1) + \hat{v}(k)] \quad (3X1) \quad (3.51)$$

for  $k = 2, 3, \dots, N_s$ , where  $N_s$  is the number of samples in the signal being integrated.

### 3.3.5 Jacobian Matrices Required by the Extended Kalman Filter

Referring to the section on the Extended Kalman Filter algorithm, we see that the EKF requires some Jacobian matrices for the formulation. First, we must define the nonlinear functions  $a[\underline{x}(t-1)]$ ,  $b[\underline{u}(t-1)]$  and  $c[\underline{x}(t)]$  in the general Gauss-Markov model according to the actual nonlinear functions in the Gauss-Markov model for our application [17]. By inspection, we see that:

$$a[\underline{\omega}(k)] \triangleq \underline{\omega}(k) - \tilde{\Psi}[\underline{\omega}(k)]\tilde{\Gamma}_U \underline{C}[\underline{\omega}(k)] \quad (3X1) \quad (3.52)$$

$$b[\underline{A}(k), \underline{\omega}(k)] \triangleq \tilde{\Psi}[\underline{\omega}(k)]\tilde{\Gamma}_U \underline{A}(k) \quad (3X1) \quad (3.53)$$

$$c[\underline{\omega}(k)] \triangleq \underline{\omega}(k) \quad (3X1) \quad (3.54)$$

With considerable analytical manipulation, we can show that the Jacobian matrices required for the Extended Kalman Filter are written as follows:

$$\tilde{\mathbb{A}}[\underline{\hat{\omega}}(k+1|k)] \triangleq \left. \frac{\partial a[\underline{\omega}(k)]}{\partial \underline{\omega}(k)} \right|_{\underline{\omega}(k)=\underline{\hat{\omega}}(k+1|k)} \quad (3.55)$$

$$= \tilde{I} - \tilde{G}^T[\underline{\hat{\omega}}(k+1|k)]\tilde{\Psi}^T[\underline{\hat{\omega}}(k+1|k)] \quad (3X3) \quad (3.56)$$

$$\tilde{\mathbb{B}}[\underline{\hat{\omega}}(k+1|k)] \triangleq \left. \frac{\partial b[\underline{A}(k), \underline{\omega}(k)]}{\partial \underline{A}(k)} \right|_{\underline{A}(k)} = \tilde{\Gamma}_U^T \tilde{\Psi}^T[\underline{\hat{\omega}}(k+1|k)] \quad (6X3) \quad (3.57)$$

$$\tilde{\mathbb{C}}[\underline{\hat{\omega}}(k+1|k)] \triangleq \left. \frac{\partial c[\underline{\omega}(k)]}{\partial \underline{\omega}(k)} \right|_{\underline{\omega}(k)=\underline{\hat{\omega}}(k+1|k)} = \tilde{I} \quad (3X3) \quad (3.58)$$

Note that  $\tilde{\mathbb{B}}[\underline{\hat{\omega}}(k+1|k)]$  is not required for the EKF in our problem.

## Chapter 4

# The Extended Kalman Filter Algorithm

The Extended Kalman Filter (EKF) is a state-space nonlinear state estimator that provides estimates of the state vector at each time  $t$  [17, 20]. It is the optimal least squares estimator for problems of the kind we have formulated. The EKF is an extension of the Kalman Filter (linear) because it handles a *nonlinear* Gauss-Markov model. For this paper, we refer to [17] for the EKF development and present here the following summary of the key equations needed to implement it. Figure (4.1) shows a block diagram of the re-entry vehicle system and the EKF.

In this section, we use the following notation. The discrete time index is denoted  $t$ . A “hat ” above a symbol denotes an estimate (e.g.  $\hat{\underline{x}}(t)$ ). A tilde above a symbol is used to denote an error or error covariance (e.g.  $\tilde{\underline{x}}(t|t-1)$  or  $\tilde{\tilde{P}}(t|t-1)$ ). The notation  $\tilde{\underline{x}}(t|t-1)$  is read “the error in the states at time  $t$ , given data up to time  $t-1$ ”. The double tilde on  $\tilde{\tilde{P}}(t|t-1)$  indicates an error covariance matrix.

### 4.1 Discrete-Time Nonlinear Gauss-Markov Model

Given state vector  $\underline{x}(t)$ , initial state vector  $\underline{x}(0)$ , nonlinear functions  $a(\cdot)$  and  $b(\cdot)$ , system input vector  $\underline{u}(t)$  and process noise vector  $\underline{W}(t)$ , we can write the state propagation model as follows:

$$\underline{x}(t) = a[\underline{x}(t-1)] + b[\underline{u}(t-1)] + \underline{W}(t) \quad \text{State propagation} \quad (4.1)$$



Given the system output measurement vector  $\underline{y}(t)$ , nonlinear function  $c(\cdot)$ , and measurement noise vector  $\underline{V}(t)$ , we can write the measurement propagation equation as follows:

$$\underline{y}(t) = c[\underline{x}(t)] + \underline{V}(t) \quad \text{Measurement propagation} \quad (4.2)$$

where  $\underline{W}(t)$  and  $\underline{V}(t)$  are zero-mean white Gaussian noise sequences with covariances  $\tilde{R}_W$  and  $\tilde{R}_V$ ,  $\underline{W}(t) \sim N[0, \tilde{R}_W]$  and  $\underline{V}(t) \sim N[0, \tilde{R}_V]$ . When appropriate, we can modify this Gauss-Markov model to handle colored noise [25, 17].

## 4.2 Discrete-Time Extended Kalman Filter Algorithm

Given the nonlinear Gauss-Markov model above, the discrete-time Extended Kalman Filter algorithm is summarized as follows [17]:

### Prediction

$$\hat{\underline{x}}(t|t-1) = a[\hat{\underline{x}}(t-1|t-1)] + b[\underline{u}(t-1)] \quad \text{State prediction} \quad (4.3)$$

$$\tilde{\tilde{P}}(t|t-1) = \tilde{\tilde{A}}[\hat{\underline{x}}(t|t-1)]\tilde{\tilde{P}}(t-1|t-1)\tilde{\tilde{A}}^T[\hat{\underline{x}}(t|t-1)] + \tilde{R}_W(t-1) \quad \text{Covariance prediction} \quad (4.4)$$

### Innovation

$$\underline{e}(t) = \underline{y}(t) - \hat{\underline{y}}(t|t-1) = \underline{y}(t) - c[\hat{\underline{x}}(t|t-1)] \quad \text{Innovation} \quad (4.5)$$

$$\tilde{R}_e(t) = \tilde{\mathbf{C}}[\hat{\underline{x}}(t|t-1)]\tilde{\tilde{P}}(t|t-1)\tilde{\mathbf{C}}^T[\hat{\underline{x}}(t|t-1)] + \tilde{R}_V(t) \quad \text{Innovation covariance} \quad (4.6)$$

### Gain

$$\tilde{K}(t) = \tilde{\tilde{P}}(t|t-1)\tilde{\mathbf{C}}^T[\hat{\underline{x}}(t|t-1)]\tilde{R}_e^{-1}(t) \quad \text{Kalman Gain} \quad (4.7)$$

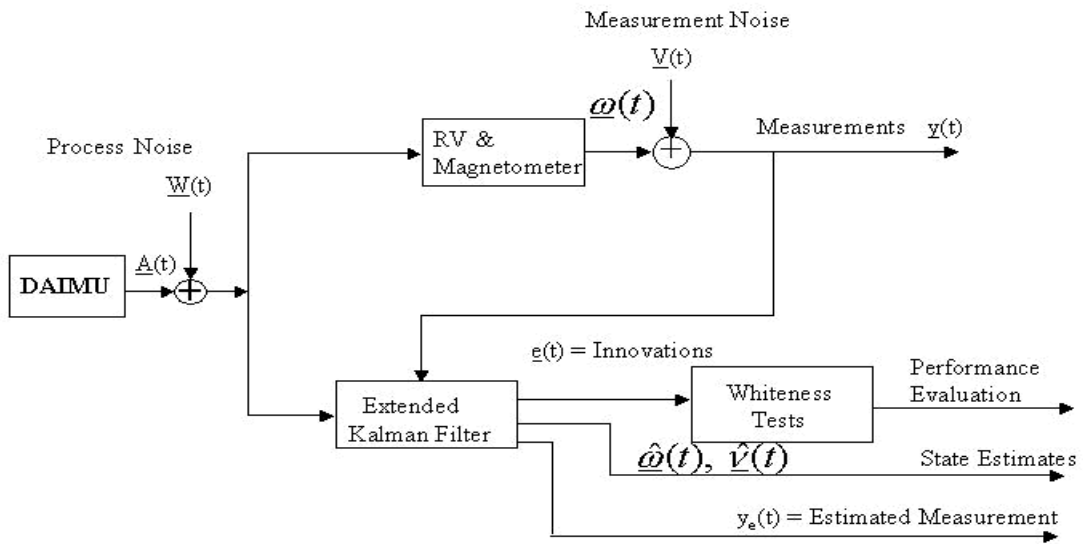


Figure 4.1: Block diagram depicting the re-entry vehicle kinematics and the Extended Kalman Filter

### Correction

$$\hat{\underline{x}}(t|t) = \hat{\underline{x}}(t|t-1) + \tilde{K}(t)\underline{e}(t) \quad \text{State correction} \quad (4.8)$$

$$\tilde{P}(t|t) = \{\tilde{I} - \tilde{K}(t)\tilde{C}[\hat{\underline{x}}(t|t-1)]\}\tilde{P}(t|t-1) \quad \text{Covariance correction} \quad (4.9)$$

### Initial Conditions

$$\hat{\underline{x}}(0|0) \text{ and } \tilde{P}(0|0)$$

### Jacobian Matrices

$$\tilde{A} \triangleq \left. \frac{\partial a[\underline{x}(t-1)]}{\partial \underline{x}(t-1)} \right|_{\underline{x}(t-1)=\hat{\underline{x}}(t|t-1)} \quad (4.10)$$

$$\tilde{B} \triangleq \left. \frac{\partial b[\underline{u}(t-1)]}{\partial \underline{u}(t-1)} \right|_{\underline{u}(t-1)} \quad (4.11)$$

$$\tilde{C} \triangleq \left. \frac{\partial c[\underline{x}(t)]}{\partial \underline{x}(t)} \right|_{\underline{x}(t)=\hat{\underline{x}}(t|t)} \quad (4.12)$$

## 4.3 Performance Measures for the EKF (Tuning)

We now summarize methods for evaluating the performance of the Kalman filter. When the Kalman filter is “tuned,” it provides an optimal or minimum error estimate of the state vector. The innovations sequence is useful for evaluating performance. A necessary and sufficient condition for the Kalman filter to be optimal is that the innovations sequence is zero-mean and white [29, 24, 17].

### 4.3.1 Zero-Mean Test on the Innovations $\underline{e}(t)$

If we assume that the innovations sequence is ergodic and Gaussian, then we can use the sample mean as the test statistic in a zero-mean hypothesis test. The mean for the  $i$ th component of  $\underline{e}(t)$  is given by:

$$\hat{m}_e(i) = \frac{1}{N} \sum_{t=1}^N e_i(t) \quad \text{for } i = 1, 2, \dots, p \quad (4.13)$$

where  $\hat{m}_e(i) \sim N(m_e, R_e(i)/N)$ ,  $p$  is the number of measurements (the number of components in  $\underline{e}(t)$ ) and  $N$  is the number of samples in the innovations sequence. The hypotheses  $H_0$  and  $H_1$  in the hypothesis test are the following:

$$H_0 : m_e = 0 \quad (4.14)$$

$$H_1 : m_e \neq 0 \quad (4.15)$$

At the  $\alpha$  significance level, the probability of rejecting the null hypothesis  $H_0$  is given by:

$$P \left( \left| \frac{\hat{m}_e(i) - m_e(i)}{\sqrt{R_e(i)/N}} \right| > \left| \frac{\tau_i - m_e(i)}{\sqrt{R_e(i)/N}} \right| \right) = \alpha \quad (4.16)$$

where  $\hat{R}_e(i)$  is the sample variance (assuming ergodicity) given by:

$$\hat{R}_e(i) = \frac{1}{N} \sum_{t=1}^N e^2(t) \quad (4.17)$$

and  $\tau_i$  is the threshold in the hypothesis test. Using a 5 % significance level ( $\alpha = .05$ ), the threshold is

$$\tau_i = 1.96 \sqrt{\frac{\hat{R}_e(i)}{N}} \quad (4.18)$$

Therefore, using the definitions above, the zero-mean hypothesis test on each component innovation  $e_i$  is given by:

$$\hat{m}_e(i) \underset{H_0}{\overset{H_1}{\geq}} \tau_i \quad (4.19)$$

In practice, we implement this test by plotting the innovations time series  $e_i(t)$  along with an overlay of the positive and negative threshold value on the same plot. We then count the number of innovations samples that exceed the threshold, divide that by the number of samples in the time series (N), and compare that fraction with the significance level ( $\alpha$ ) to decide whether or not the innovations can be deemed “white.” Of course, the test has limited value unless the data are ergodic and Gaussian.

### 4.3.2 Innovations Whiteness Test

The “whiteness” of the innovations is a measure of how well the Kalman filter is tuned. Recall that a discrete-time stochastic process is white if its autocorrelation function is a Kronecker delta [25] at zero lag. This suggests a practical statistical hypothesis test for whiteness. Assuming ergodicity, we develop a test based upon the normalized sample autocovariance function of the innovations sequence as our test statistic:

$$\hat{\rho}_e(i, k) = \frac{\hat{R}_e(i, k)}{\hat{R}_e(i)} \quad (4.20)$$

where the  $i$ th component innovations covariance is given by:

$$\hat{R}_e(i, k) = \frac{1}{N} \sum_{t=k+1}^N [e_i(t) - \hat{m}_e(i)][e_i(t+k) - \hat{m}_e(i)] \quad (4.21)$$

Note that  $i$  is the index on the number of measurements ( $i = 1, 2, \dots, p$ ) and  $k$  is the correlation lag index. For this test,  $N$  denotes the number of samples in a data window over which the covariance is calculated, so  $k = 1, 2, \dots, N$ . Usually, we let  $N$  be the number of samples in the innovations time series. Note that we sum from  $t = k + 1$  to  $N$  to avoid the first sample (at zero lag), which should equal one (the Kronecker delta) when we apply the hypothesis test described next.

Asymptotically, for large  $N$ , it can be shown [32] that the test statistic is Gaussian  $\hat{\rho}_e(i, k) \sim N[0, 1/N]$ ; therefore, the 95% confidence interval estimate of  $\hat{\rho}_e(i, k)$  is given by:

$$I_{\rho_e} = \hat{\rho}_e(i, k) \pm \frac{1.96}{\sqrt{N}} \quad (N > 30) \quad (4.22)$$

Hence under the null hypothesis ( $e_i(t)$  is white),  $\hat{\rho}_e(i, k)$  must lie within the interval  $I_{\rho_e}$  95 percent of the time to accept  $H_0$ ; i.e., to declare that the component innovation is white.

In practice, we implement this test by plotting the normalized autocovariance  $\hat{\rho}_e(i, k)$  over  $N$  lags ( $N > 30$ ), with the threshold  $1.96/\sqrt{N}$  overlayed on the same plot. We count the number of samples that exceed the threshold, divide it by  $N$ , and compare that fraction to the significance level (.05 because we use a 95 percent confidence interval) to decide whether the innovations are white.

### 4.3.3 Weighted Sum Squared Residual (WSSR)

The whiteness test above has great value for evaluating the whiteness of one innovations component. For systems with multiple measurements, the analysis of all the innovations separately can be cumbersome and may not provide sufficient information about the performance of the overall system. The Weighted Sum Squared Residual (WSSR) provides a method for whiteness testing of the overall state estimate by aggregating innovations vector information  $\underline{e}(t)$  after a finite time window of length  $N$  into a single scalar test statistic. It can be shown that the WSSR is related to an estimate of the normalized innovations variance [34]. We define the WSSR as a scalar test statistic  $\rho$  as follows:

$$\rho(l) = \sum_{k=l-N+1}^l \underline{e}^T(k) R_e^{-1}(k) \underline{e}(k) \quad \text{for } l \geq N \quad (\text{scalar}) \quad (4.23)$$

Note that the WSSR is evaluated only for  $l \geq N$ , because we wish to inspect the covariance at lags *after* which the transient has settled down to a reasonable “steady-state.” The hypothesis test for overall whiteness becomes

$$\rho(l) \underset{H_0}{\overset{H_1}{\geq}} \tau \quad (\tau = \text{decision threshold}) \quad (4.24)$$

Under the null hypothesis,  $\rho(l) \sim \chi^2(Np)$ . However, for  $Np \geq 30$ ,  $\rho(l)$  is approximately  $N(Np, 2Np)$  (see Anderson [31] or Schweppe [33] for details. At the  $\alpha$  significance level, the probability of rejecting the null hypothesis is

$$P\left(\left|\frac{\rho(l)-Np}{\sqrt{2Np}}\right| > \left|\frac{\tau-Np}{\sqrt{2Np}}\right|\right) = \alpha \quad (4.25)$$

For a significance level of  $\alpha = .05$ , the threshold is

$$\tau = Np + 1.96\sqrt{2Np} \quad (4.26)$$

where  $p$  is the number of measurements and  $N$  is the number of covariance lag samples after which we wish to evaluate the WSSR. Note that the WSSR test can be used for both stationary and nonstationary processes. Note also that the value of  $N$  can be adjusted in the WSSR test.

In practice, we implement the WSSR test by plotting the WSSR for lags beyond  $N$  and overlaying the threshold  $\tau$  on the plot. We count the number of WSSR samples that exceed the threshold, divide that by the number of WSSR samples calculated and compare that fraction with the significance level  $\alpha$  to decide whether or not the aggregated innovation information can be deemed white.

#### 4.3.4 “Reasonableness Tests ” on Covariances

The covariances estimated by the Kalman filter algorithm can be used for tests of how well the filter is tuned. When the filter reaches steady state (the process is stationary),  $\tilde{P}$  is a constant, so the estimates can be compared to ensure that they are reasonable. We can test for:

$$\hat{R}_e(i) \approx R_e(i) \quad \text{and} \quad \hat{\tilde{P}}(i) \approx \tilde{P}(i) \quad (4.27)$$

#### 4.3.5 Tests When the True State Vector is Known

When the true state vector  $X_{true}$  is known, we can plot the “two-sigma” statistical bounds  $\pm 2\sqrt{R_{e_i}}$  about the estimation error sequence  $\{\tilde{x}(t|t)\}$  and  $\pm 2\sqrt{\tilde{P}_i(t|t)}$  about the innovations sequence  $\{e_i(t)\}$ . If the covariance estimates of the filter are reasonable, then 95 percent of the sequence samples should lie within the two-sigma bounds. See [17] for a comprehensive discussion of Kalman filter tuning and performance measurement.

## Chapter 5

# Simulation Experiment

For algorithm testing purposes, we have constructed an artificial problem that has some of the general characteristics of the DAIMU problem. This problem allows us to analyze the performance of the modeling and estimation algorithms before we apply actual experimental data to them. The algorithms and experiments are implemented in MATLAB [38] software.

### 5.1 Choices for the Velocity and Acceleration Vectors $\underline{B}(t)$ and $\underline{\dot{B}}(t)$

Recall that we can write the vector ordinary differential equation (ODE) describing the kinematics of the system as follows:

$$\underline{A}(t) = \tilde{T}\underline{\dot{B}}(t) + \underline{C}[\underline{B}(t)] \quad (5.1)$$

where  $\tilde{T}$  is a 6X6 constant matrix designed to be nonsingular and determined by the spatial configuration of the accelerometer array in the DAIMU. We use simple sinusoids and parabolic functions for simulation. Recall the following definitions from equation (3.6):

$$\underline{B}(t) = [\omega_x(t) \ \omega_y(t) \ \omega_z(t) \ \nu_x(t) \ \nu_y(t) \ \nu_z(t)]^T \quad (6X1) \quad (5.2)$$

$$\underline{\dot{B}}(t) = [\dot{\omega}_x(t) \ \dot{\omega}_y(t) \ \dot{\omega}_z(t) \ \dot{\nu}_x(t) \ \dot{\nu}_y(t) \ \dot{\nu}_z(t)]^T \quad (6X1) \quad (5.3)$$



The individual angular and linear velocities (along with their derivatives and initial values) are specified as follows. Note that the symbol  $\Omega$  used in this section does not have the same meaning(s) it did in previous sections.

$$\omega_x(t) = \sin(\Omega t), \quad \dot{\omega}_x(t) = \Omega \cos(\Omega t) \quad (5.4)$$

$$\omega_y(t) = \cos(\Omega t), \quad \dot{\omega}_y(t) = -\Omega \sin(\Omega t) \quad (5.5)$$

$$\omega_z(t) = \omega_0 + \alpha t^2, \quad \dot{\omega}_z(t) = 2\alpha t \quad (5.6)$$

$$\nu_x(t) = \cos(\beta t), \quad \dot{\nu}_x(t) = -\beta \sin(\beta t) \quad (5.7)$$

$$\nu_y(t) = \sin(\beta t), \quad \dot{\nu}_y(t) = \beta \cos(\beta t) \quad (5.8)$$

$$\nu_z(t) = \nu_0 - \gamma t^2, \quad \dot{\nu}_z(t) = -2\gamma t \quad (5.9)$$

$$\omega_x(0) = 0, \quad \dot{\omega}_x(0) = \Omega \quad (5.10)$$

$$\omega_y(0) = 1, \quad \dot{\omega}_y(0) = 0 \quad (5.11)$$

$$\omega_z(0) = 2, \quad \dot{\omega}_z(0) = 0 \quad (5.12)$$

$$\nu_x(0) = 1, \quad \dot{\nu}_x(0) = 0 \quad (5.13)$$

$$\nu_y(0) = 0, \quad \dot{\nu}_y(0) = \beta \quad (5.14)$$

$$\nu_z(0) = \nu_0, \quad \dot{\nu}_z(0) = 0 \quad (5.15)$$

$$\Omega = .3142 \quad (5.16)$$

$$\alpha = 4.e - 4 \quad (5.17)$$

$$\beta = .2356 \quad (5.18)$$

$$\omega_0 = 2 \quad (5.19)$$

$$\nu_0 = 15 \quad (5.20)$$

$$\gamma = -5.e - 4 \quad (5.21)$$

$$T_s = .1 \text{ sec. (sampling period)} \quad (5.22)$$

$$N = 1000 \text{ (Number of samples in the data record)} \quad (5.23)$$

$$T_0 = NT_s = 100 \text{ sec.} \quad (5.24)$$

$$\tilde{T} = \begin{pmatrix} 2.0076 & 180.1947 & 67.7616 & 0.8053 & -0.2165 & 0.5519 \\ 2.9223 & -31.2313 & -189.8224 & 0.8075 & 0.5839 & -0.0836 \\ 2.1828 & -148.7878 & 122.0883 & 0.8069 & -0.3675 & -0.4623 \\ 157.8485 & 107.7899 & -19.1435 & -0.0063 & -0.1659 & -0.9861 \\ 157.5631 & -72.3901 & -81.9704 & -0.0192 & -0.7674 & 0.6409 \\ 157.8026 & -37.1743 & 102.8329 & -0.0066 & 0.9371 & 0.3490 \end{pmatrix} \quad (5.25)$$

Figures (5.1) and (5.2) show the simulated  $\underline{B}(t)$  and  $\underline{\dot{B}}(t)$  vectors generated in MATLAB using the specifications above.

## 5.2 Simulate the Accelerations $\underline{A}(t)$

Here, we use the chosen definitions above to simulate the accelerations using the ODE simulator and plot them in Figure (5.3).

## 5.3 Simulate the States $\underline{\omega}(t)$ Using the Gauss-Markov Model

Using the acceleration vector  $\underline{A}(t)$  and the initial conditions, we calculate the noiseless states  $\underline{\omega}(t)$  from the Gauss-Markov model and plot them in Figure (5.4). In Figure (5.5), the states calculated by the Gauss-Markov model are overlayed on the states we defined. We see that they agree. We then add white Gaussian measurement noise and plot the noisy measurements in Figure (5.6).

## 5.4 Estimate the States $\underline{\hat{\omega}}(k)$ Using the Extended Kalman Filter

Figure (5.7) shows a plot of the estimated states  $\underline{\hat{\omega}}(k)$  from the EKF overlayed on a plot of the simulated states  $\underline{\omega}(k)$ . The corresponding estimates  $\underline{\hat{y}}(k)$  of the outputs are plotted in Figure (5.9). Figure (5.10) plots the simulated outputs  $\underline{y}(k)$  from the Gauss-Markov model overlayed on the estimated outputs  $\underline{\hat{y}}(k)$  from the EKF.

The estimated linear velocities  $\underline{\hat{v}}(k)$  are obtained by integrating the estimated  $\underline{\hat{p}}(k)$  obtained by using the  $\underline{\hat{\omega}}(k)$  from the EKF in the ODE (see the text). We plot the estimated linear velocities in Figure (5.8). We see that the linear velocity estimates are unacceptable. The numerical difficulties associated with this calculation are described in a later section.

## 5.5 Examine the Performance of the EKF

We now examine the performance of the EKF using the methods described in the text. First, we plot the innovations  $\underline{e}(k)$  from the EKF along with their

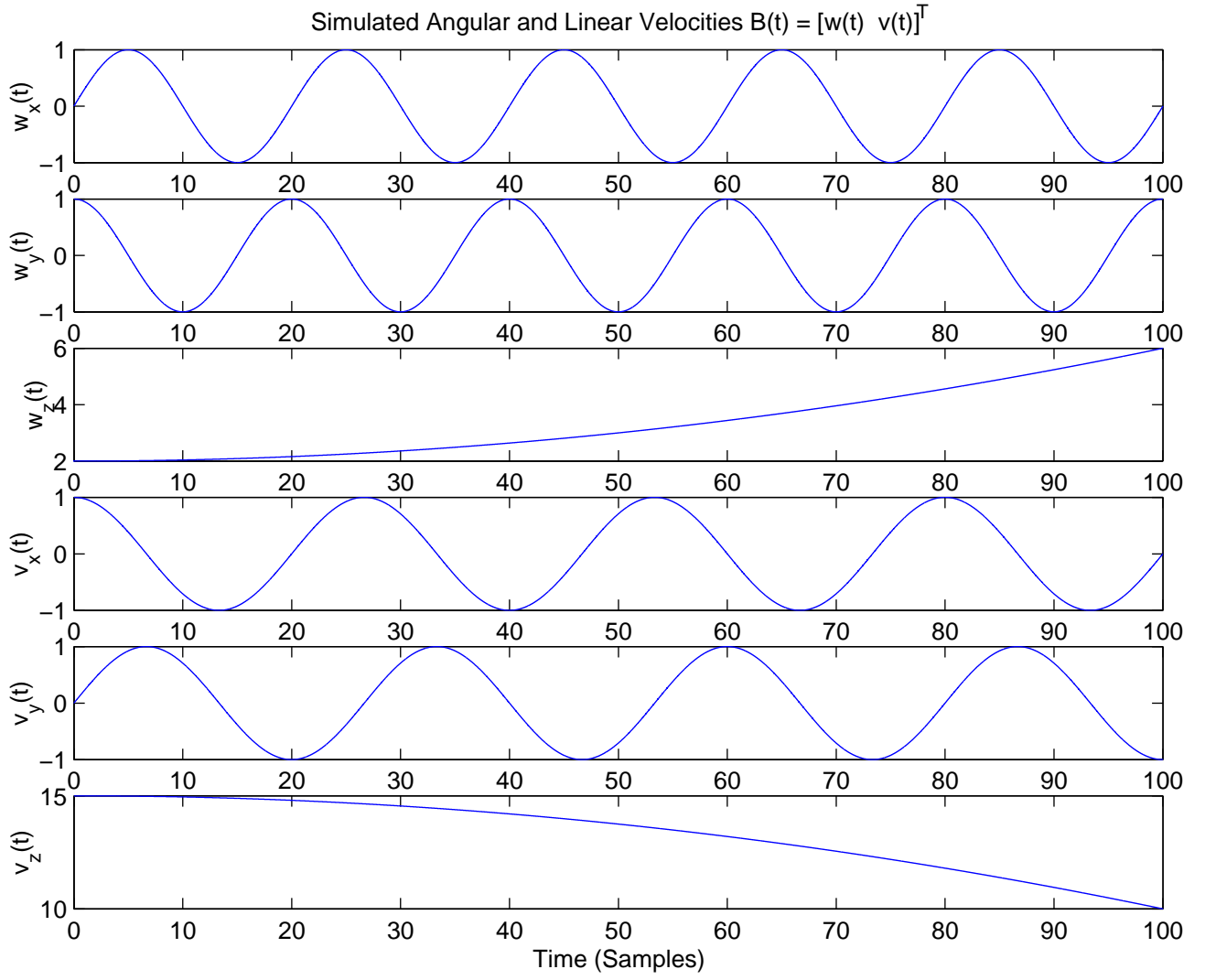


Figure 5.1: (Exp.1) Simulated angular velocity  $\underline{\omega}(k)$  and linear velocity  $\underline{v}(k)$  (F1)

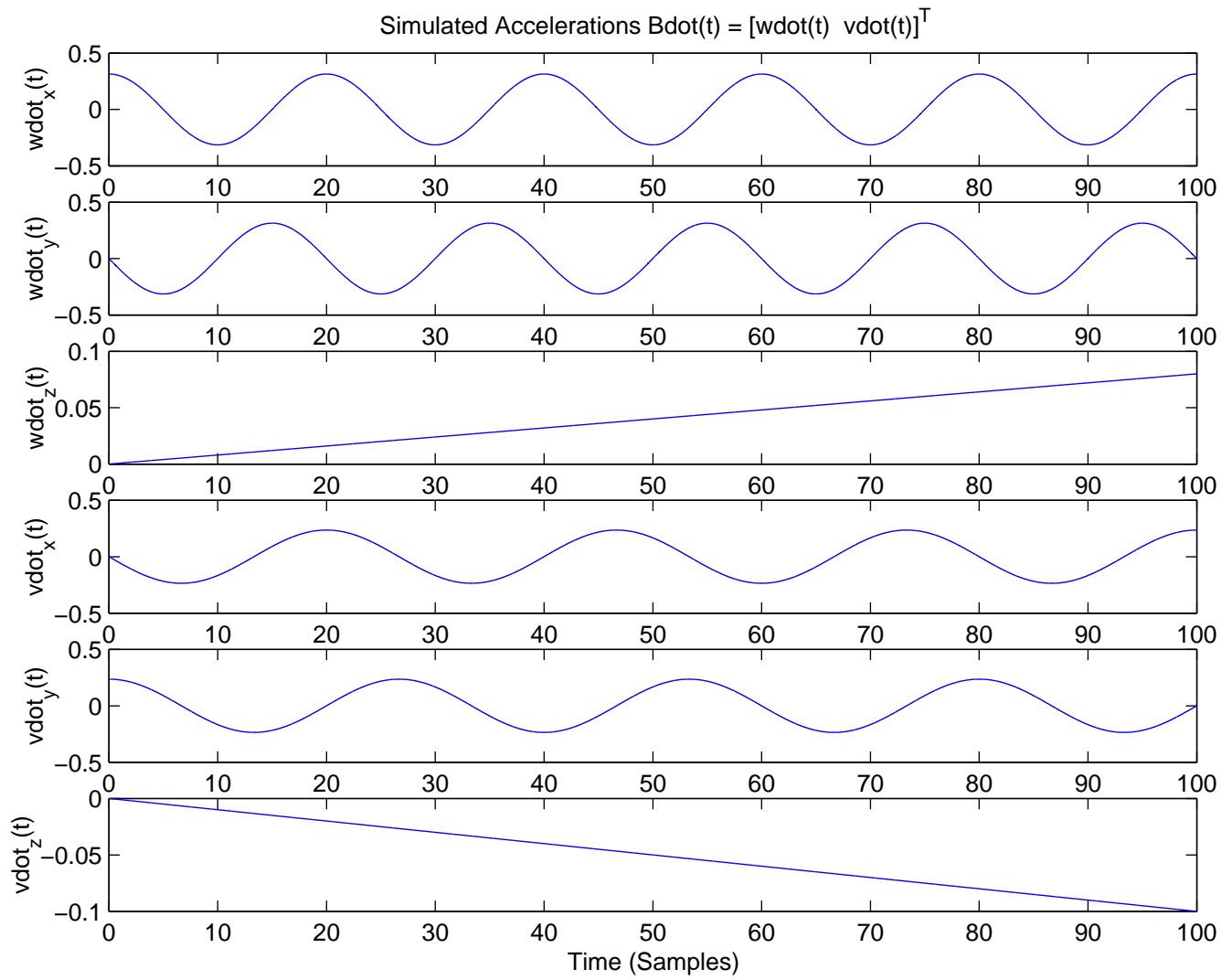


Figure 5.2: (Exp.1) Simulated Accelerations  $\dot{\underline{B}}(t)$  (F2)

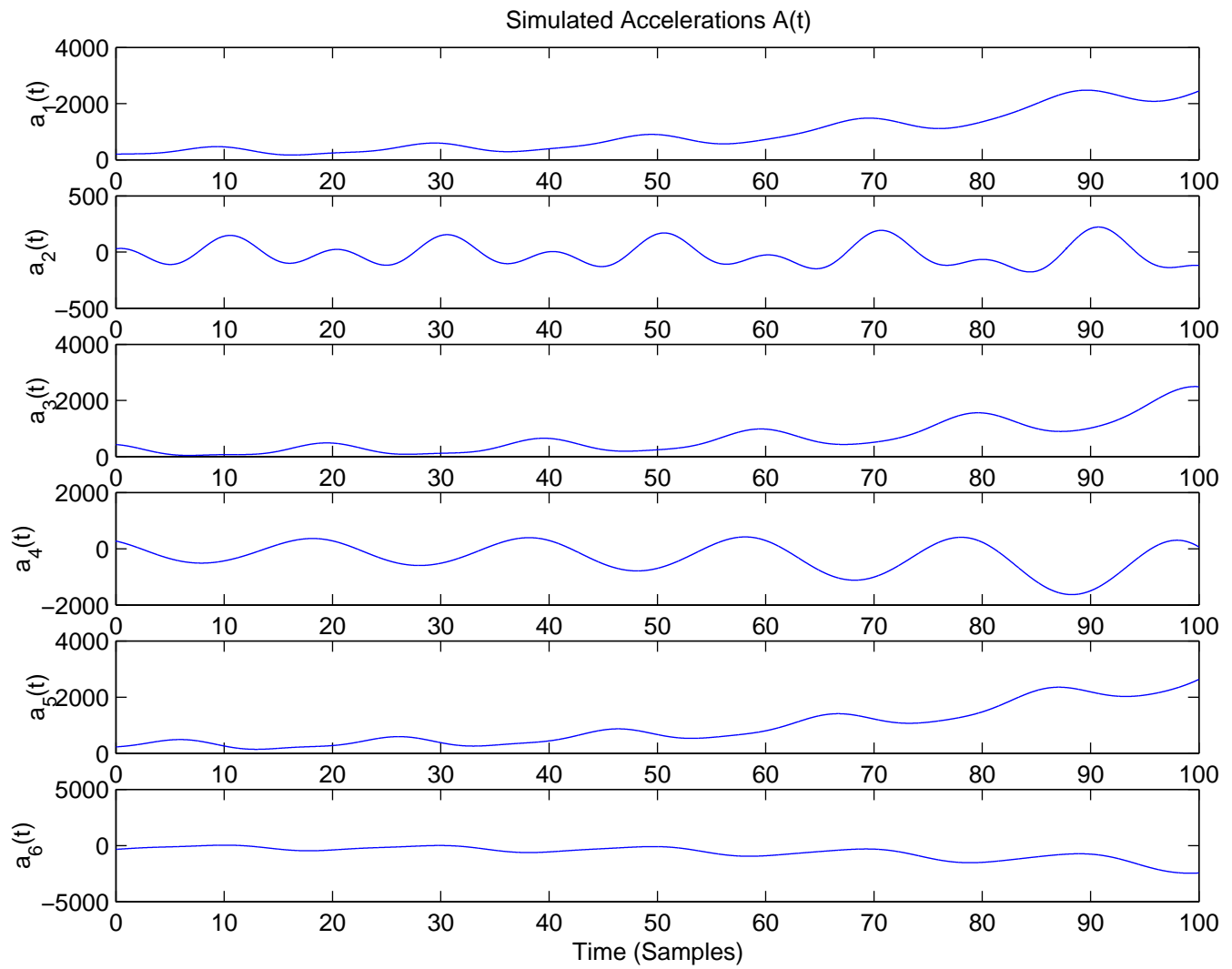


Figure 5.3: (Exp.1) Simulated Acceleration Measurements  $\underline{A}(t)$  from the ODE Simulator (F3)

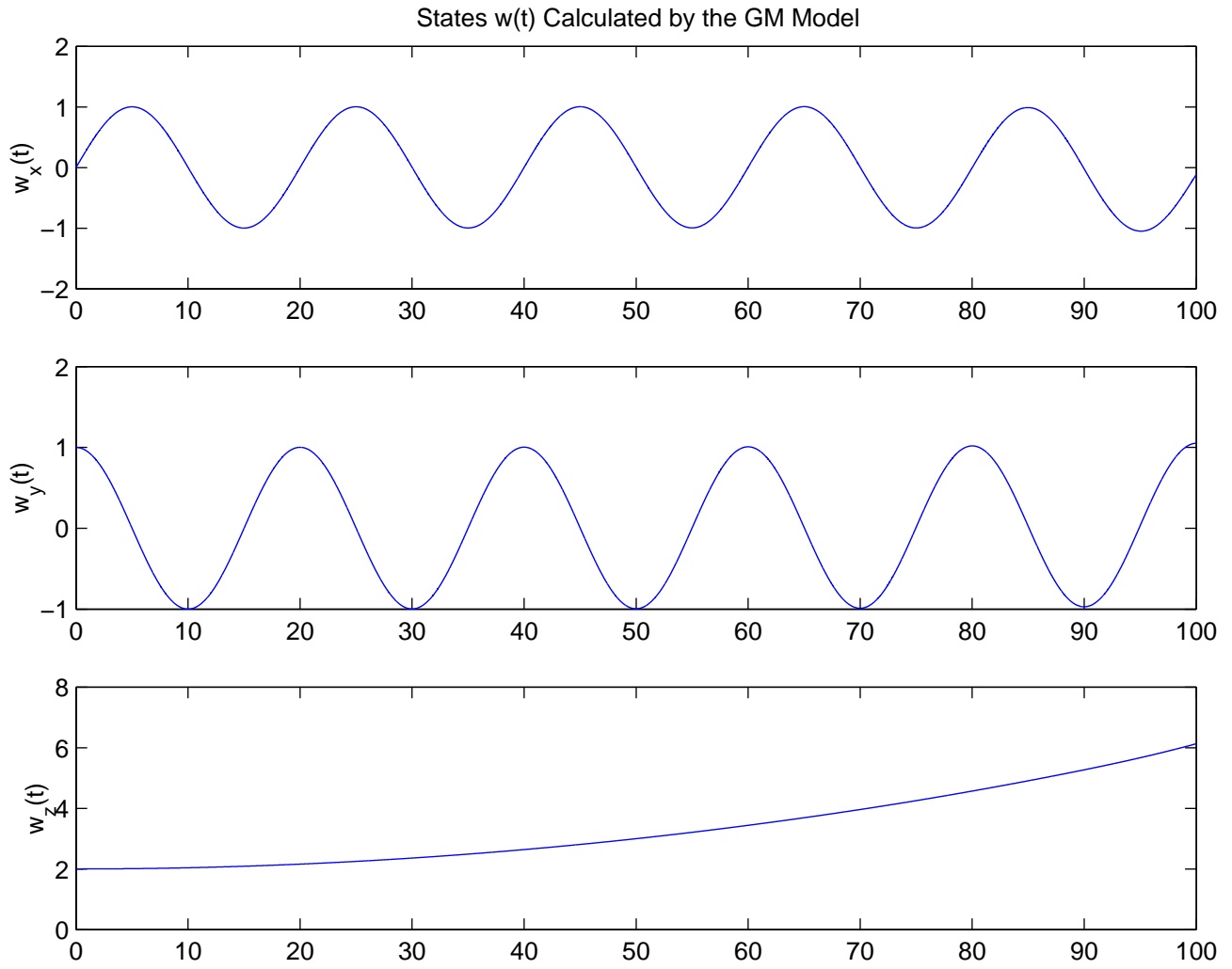


Figure 5.4: (Exp.1) States  $\underline{w}(t)$  Calculated by the Gauss-Markov Model (F22)

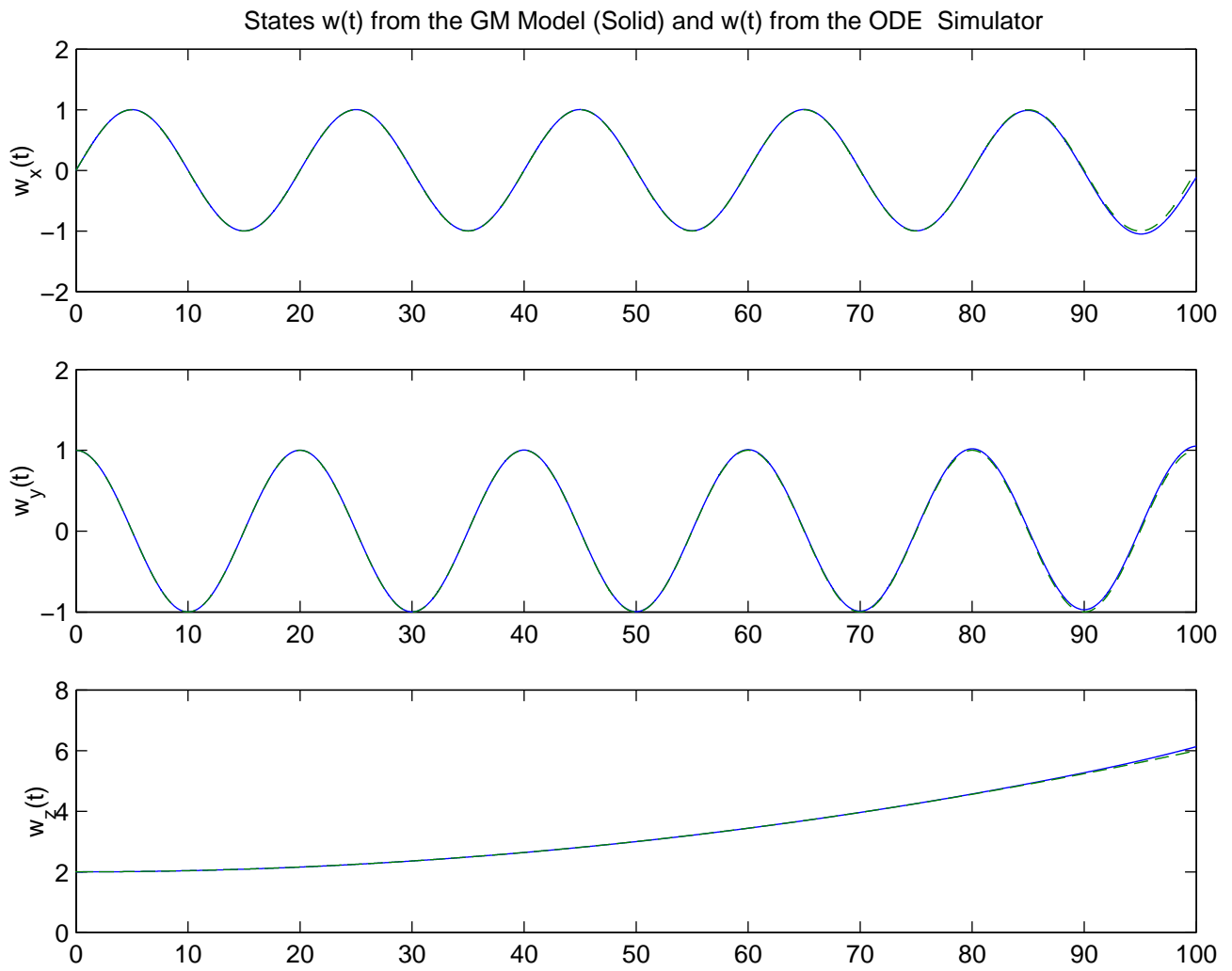


Figure 5.5: (Exp.1) States  $\underline{w}(t)$  from the Gauss-Markov Model (Solid) and from the ODE Simulator (Dashed) (F26)

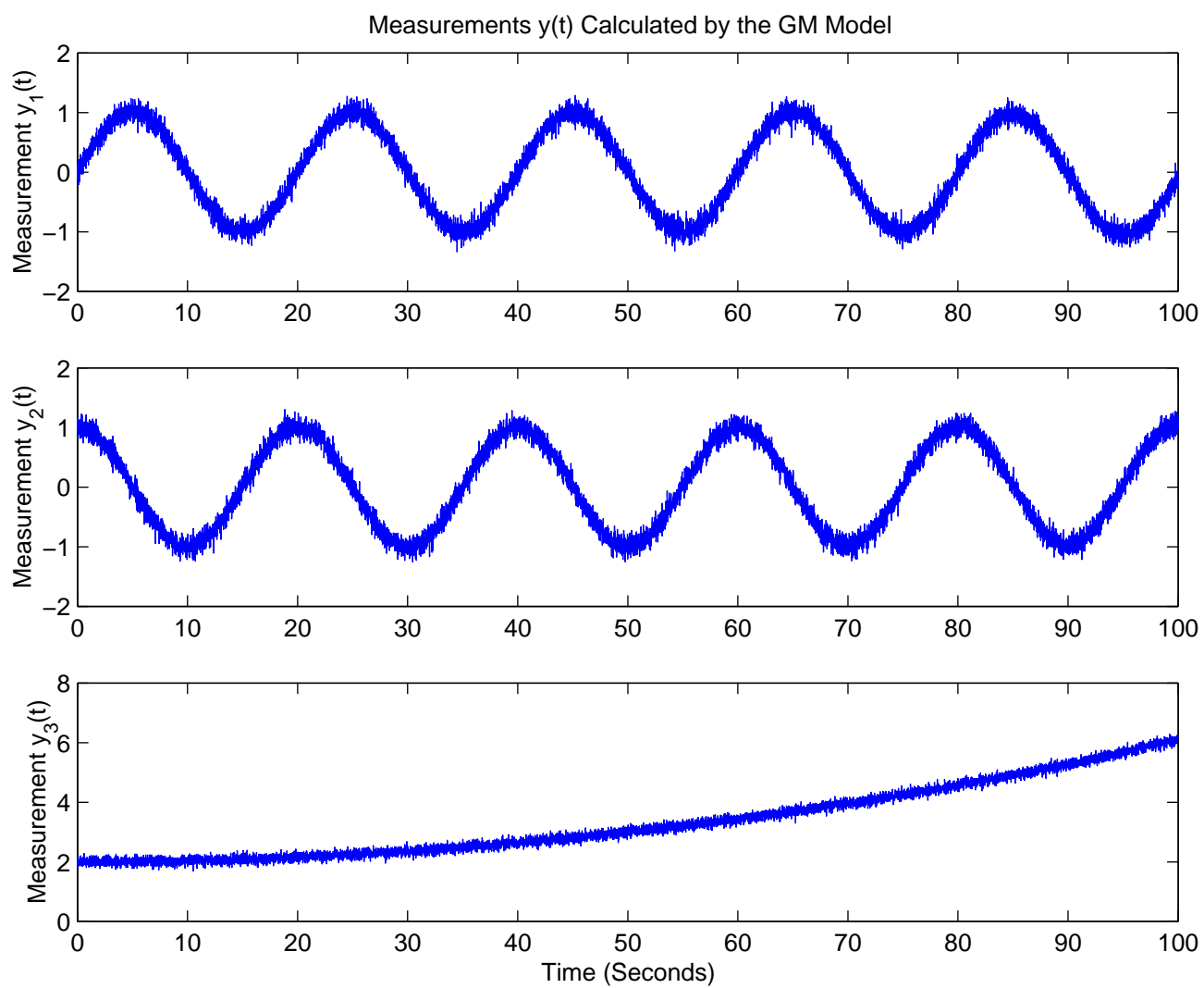


Figure 5.6: (Exp.1) Measurements  $\underline{y}(t)$  Calculated by the Gauss-Markov Model (F27)



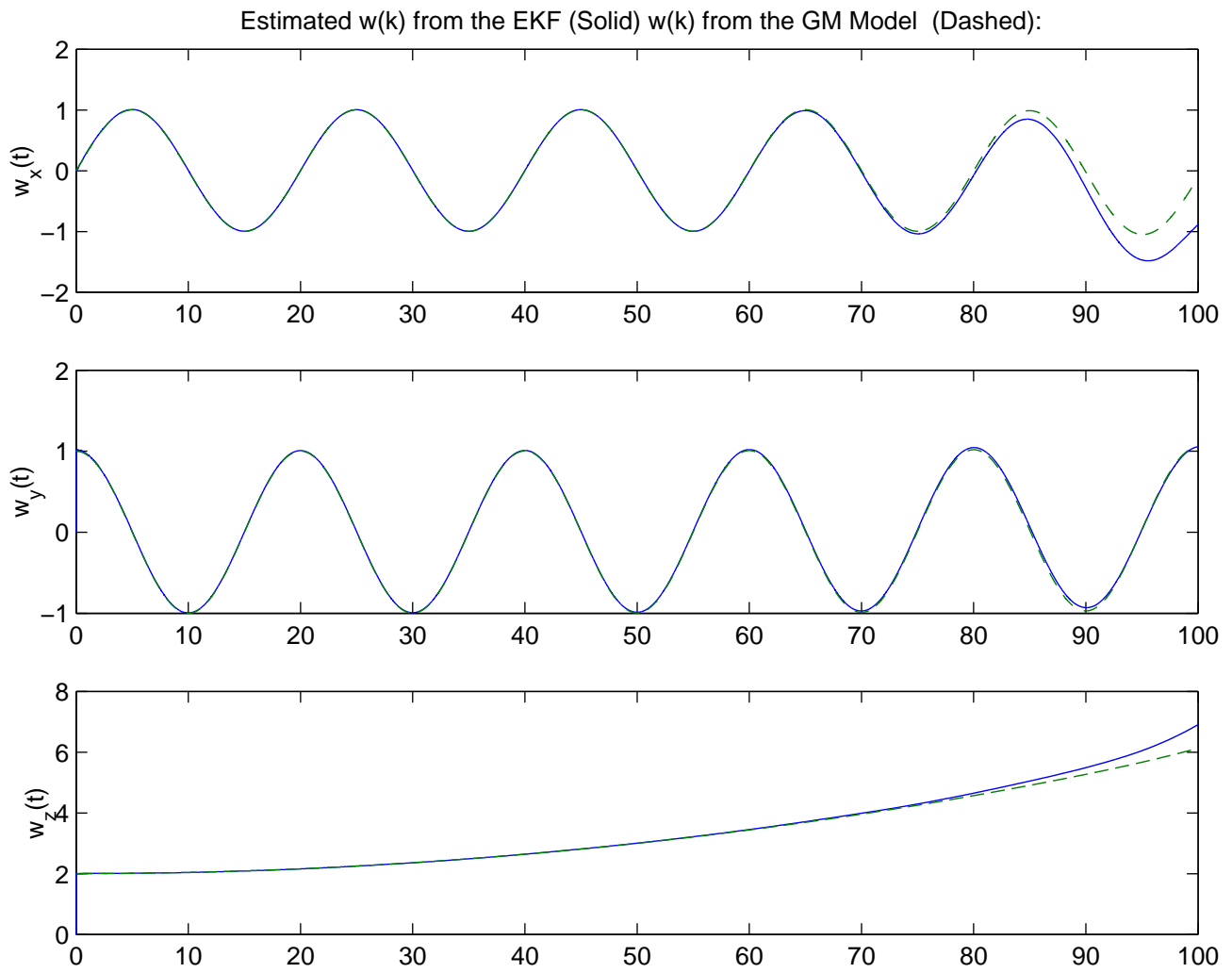


Figure 5.7: (Exp.1) Estimated states  $\hat{\underline{w}}(k)$  from the EKF (solid lines) and simulated states  $\underline{w}(k)$  from the Gauss- Markov model (dashed lines) (F38)

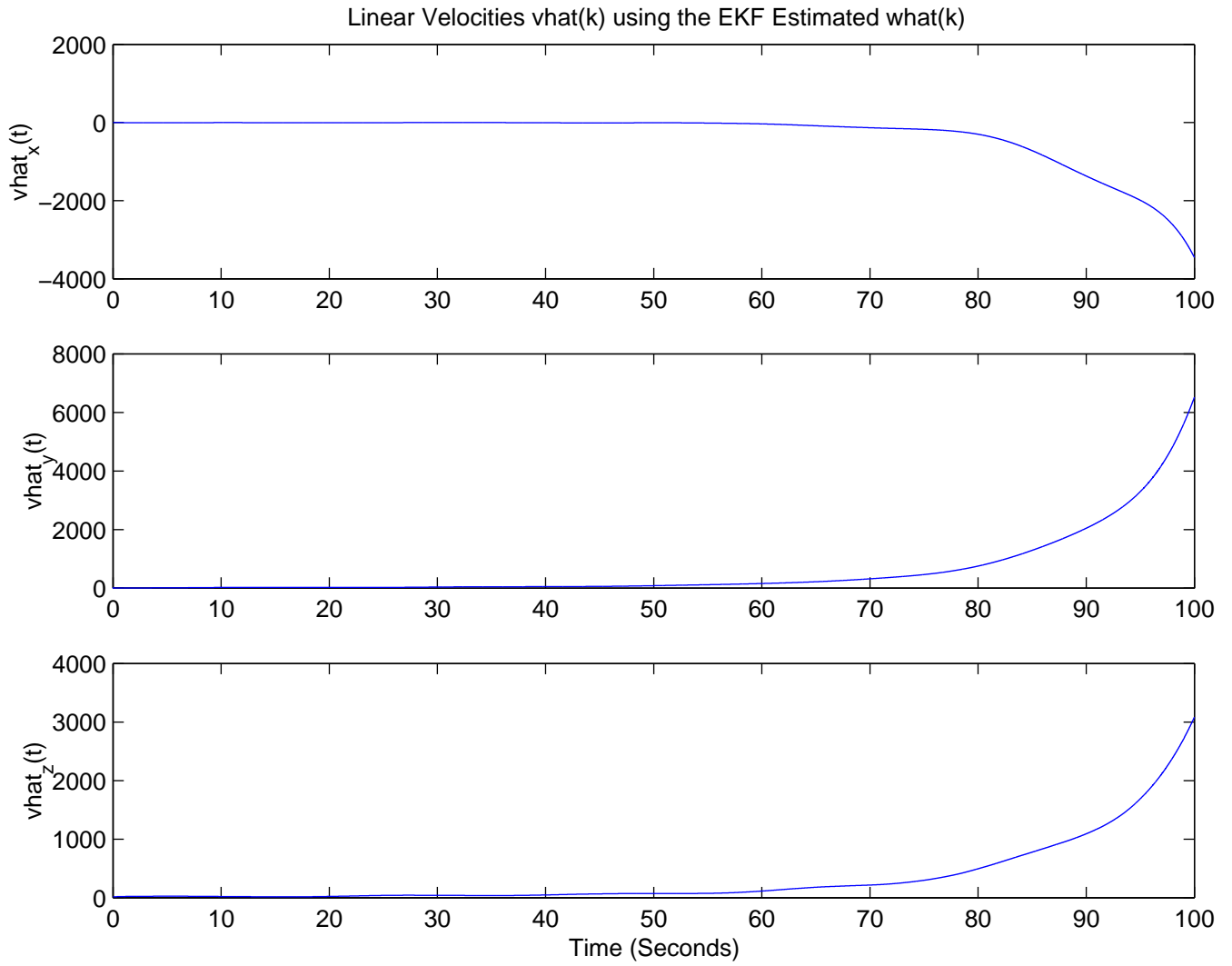


Figure 5.8: (Exp.1) Estimated linear velocities  $\hat{\underline{v}}(k)$  from quadrature on the estimated  $\hat{\underline{v}}(k)$  obtained by using the  $\hat{\underline{w}}(k)$  from the EKF in the ODE's (F49)

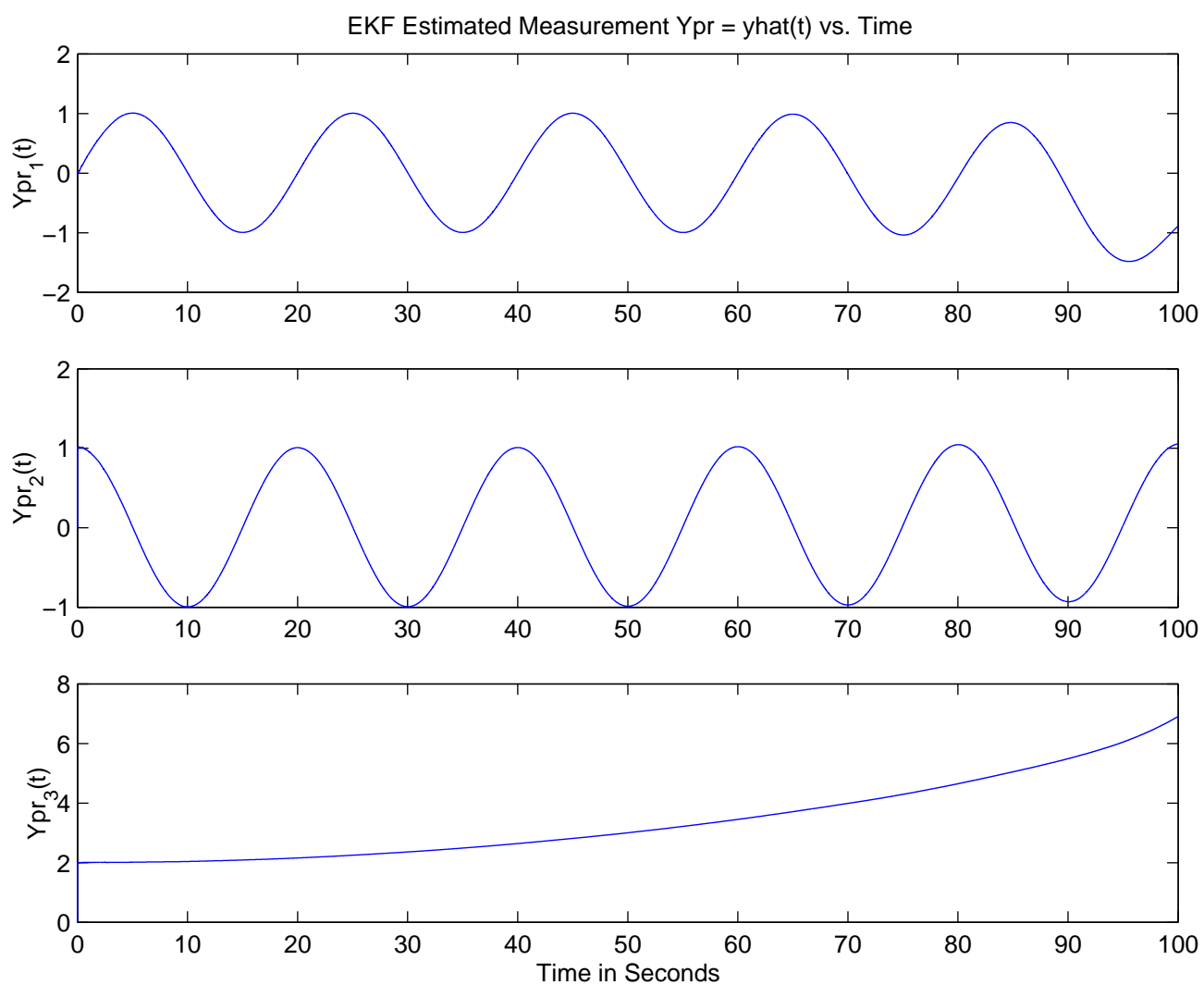


Figure 5.9: (Exp.1) Estimated outputs  $\hat{\underline{y}}(k)$  from the EKF (F41)

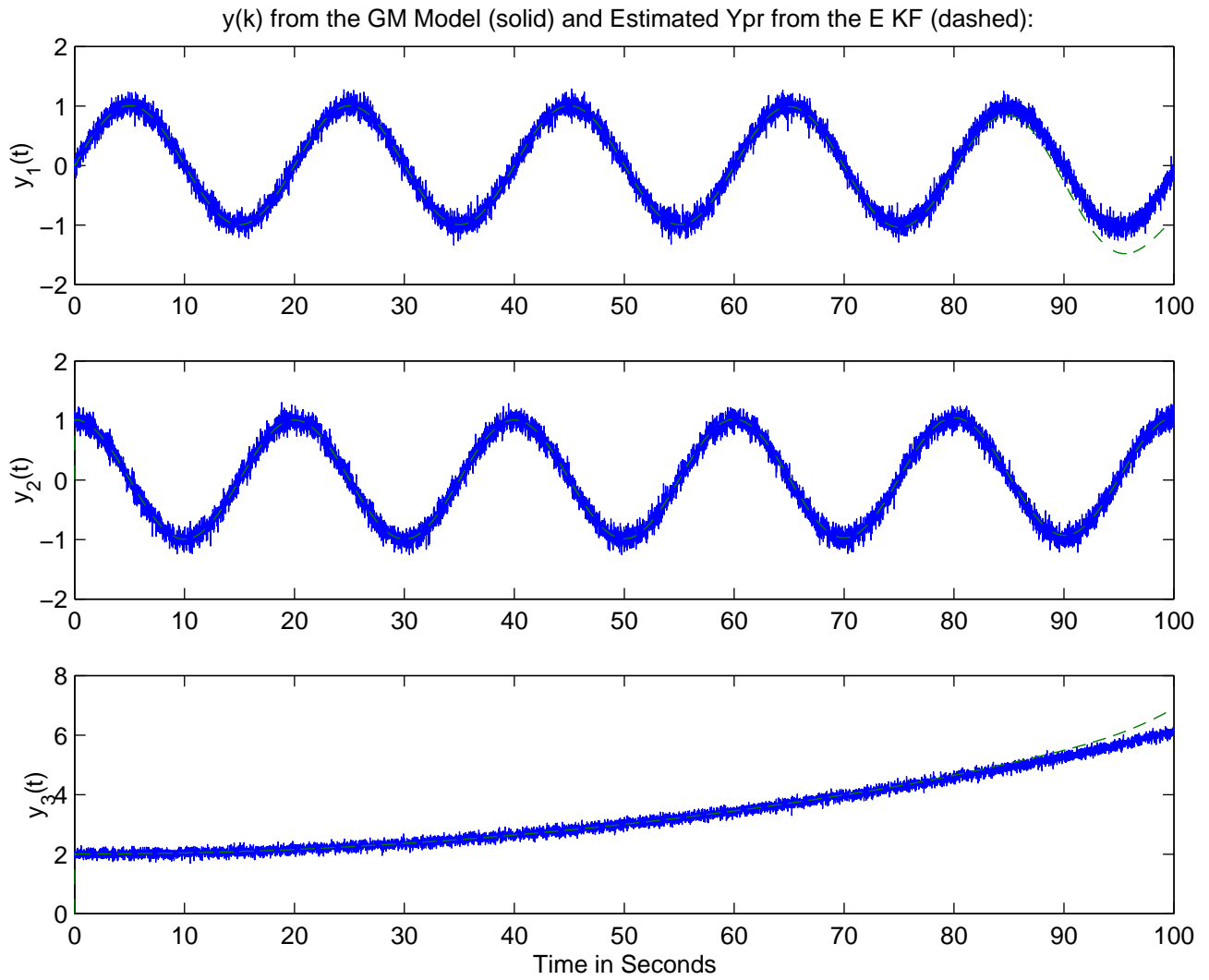


Figure 5.10: (Exp.1) Simulated outputs  $\underline{y}(k)$  from the Gauss- Markov model (solid lines) and estimated outputs  $\hat{\underline{y}}(k)$  from the EKF (dashed lines) (F42)

“two-sigma bounds ” in Figure (5.11) This plot shows that the exhibit good behavior until near the end of the data record, where their means become nonzero and their magnitudes increase with time.

Figure (5.12), Figure (5.13) and Figure (5.14) show the whiteness test plots for the three components of the innovations vector  $\underline{e}(k)$ . Figure (5.12) shows that the x-component of the innovations is not white. Even though the autocovariance converges to a value inside the 2-sigma bounds, there are too many samples outside the bounds at small lags. Figure (5.13) indicates that the y-component of the innovations comes close to meeting the whiteness criterion, because 7 percent of the samples of the autocovariance lie outside the 2-sigma bounds, but our whiteness criterion requires less than 5 percent. Figure (5.14) shows that the result for the z-component are about the same as those for the x-component. The drift at the end of the estimation data records is responsible for the lack of desired results.

Interestingly, we reach different conclusions about the EKF performance when we examine the weighted sum squared error (WSSR) in Figure (5.15) for the innovations vector  $\underline{e}(k)$ . Here, we see that the WSSR never exceeds its threshold, so by this criterion, we can declare the EKF to be tuned. Even though the numerical issues cause the WSSR to rise significantly at large lags, the overall performance remains acceptable.

## 5.6 Mathematical/Numerical Performance Issues

There are two mathematical problems associated with the DAIMU configuration, and they both are evident in this example problem: (1) The inherently unstable nature of the nonlinear ODE's (discussed earlier), and (2) The numerical problem associated with computing the difference  $\underline{A}(k) - \underline{C}[\underline{\hat{\omega}}(k)]$ . We discuss these next.

The ODE solutions contain drift/bias due to the nonlinear (quadratic) centripetal acceleration term. This effect worsens with increasing angular velocity [12]. We can see from the figures in the last section, especially Figure (5.8), that this unstable nature of the nonlinear ODE's cause the estimates to diverge near the end of the data record.

Another problem is very apparent from this analysis. Recall the vector ODE for the system kinematics:

$$\dot{\underline{B}}(t) = \tilde{T}^{-1}\{\underline{A}(t) - \underline{C}[\underline{\omega}(t)]\} \quad (6X1) \quad (5.26)$$

There are numerical difficulties associated with calculating the difference

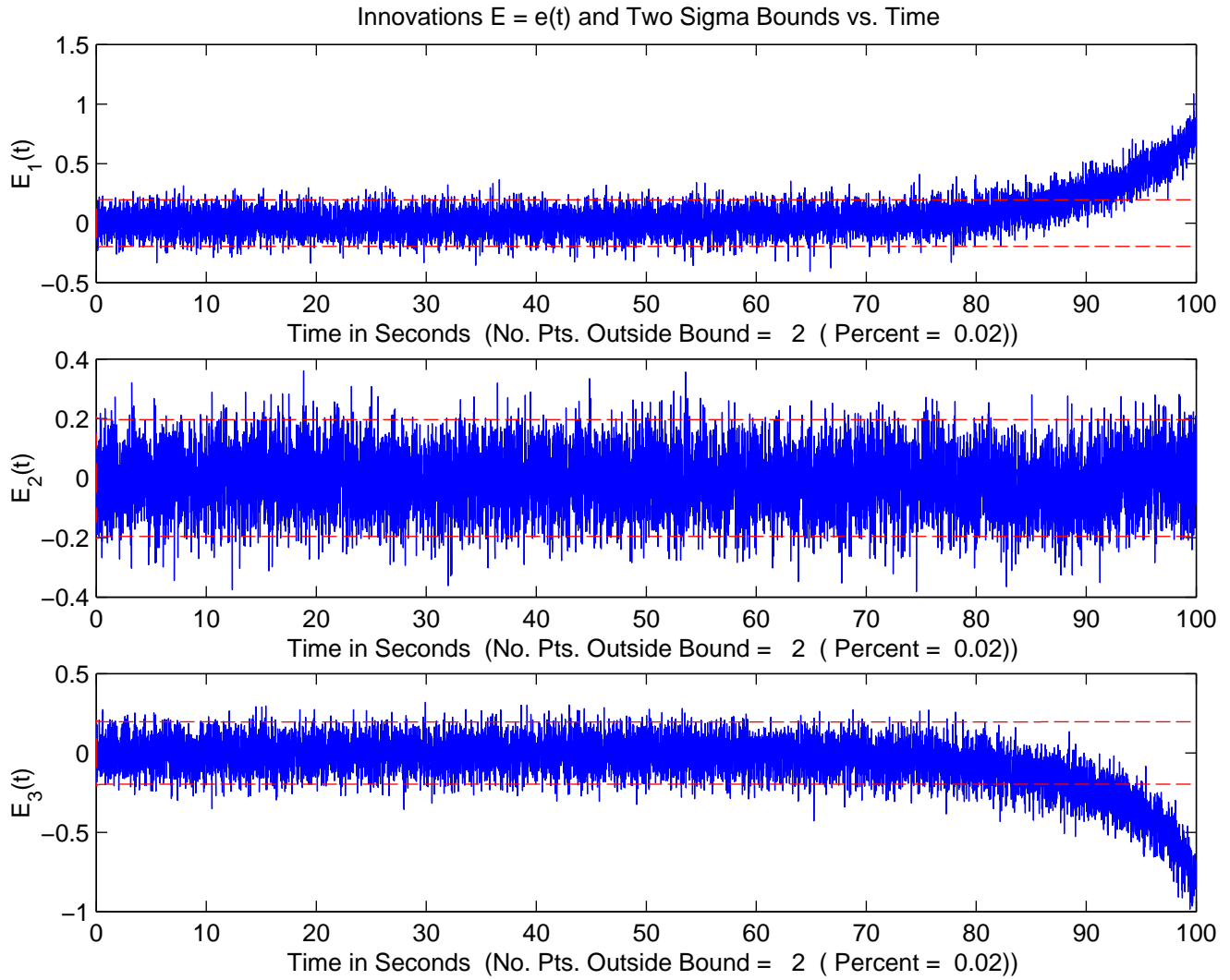


Figure 5.11: (Exp.1) Innovations  $\underline{e}(k)$  and their “two-sigma bounds” from the EKF (F43)

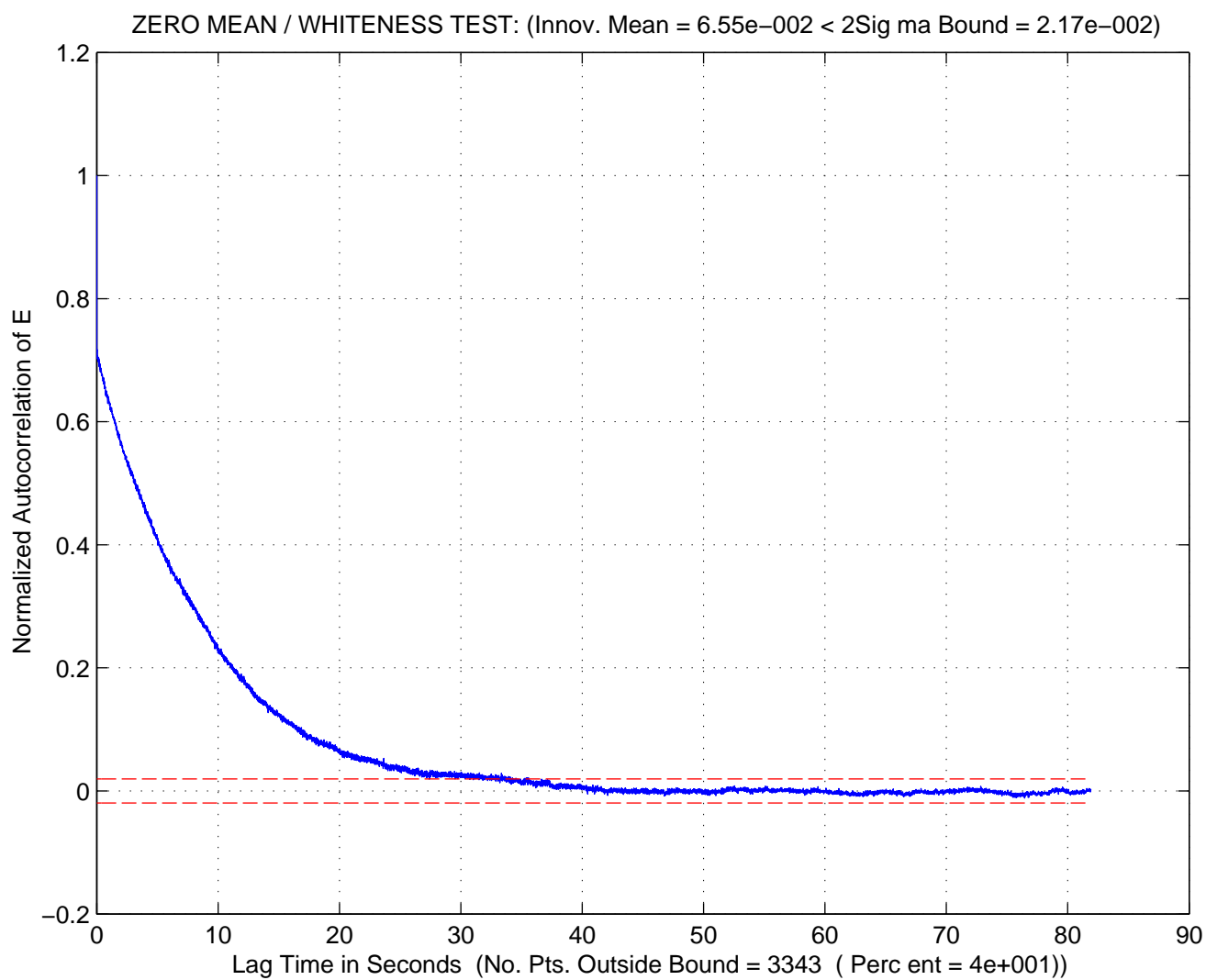


Figure 5.12: (Exp.1) Whiteness test for the x-component of the innovations  $e_x(k)$  (F44)

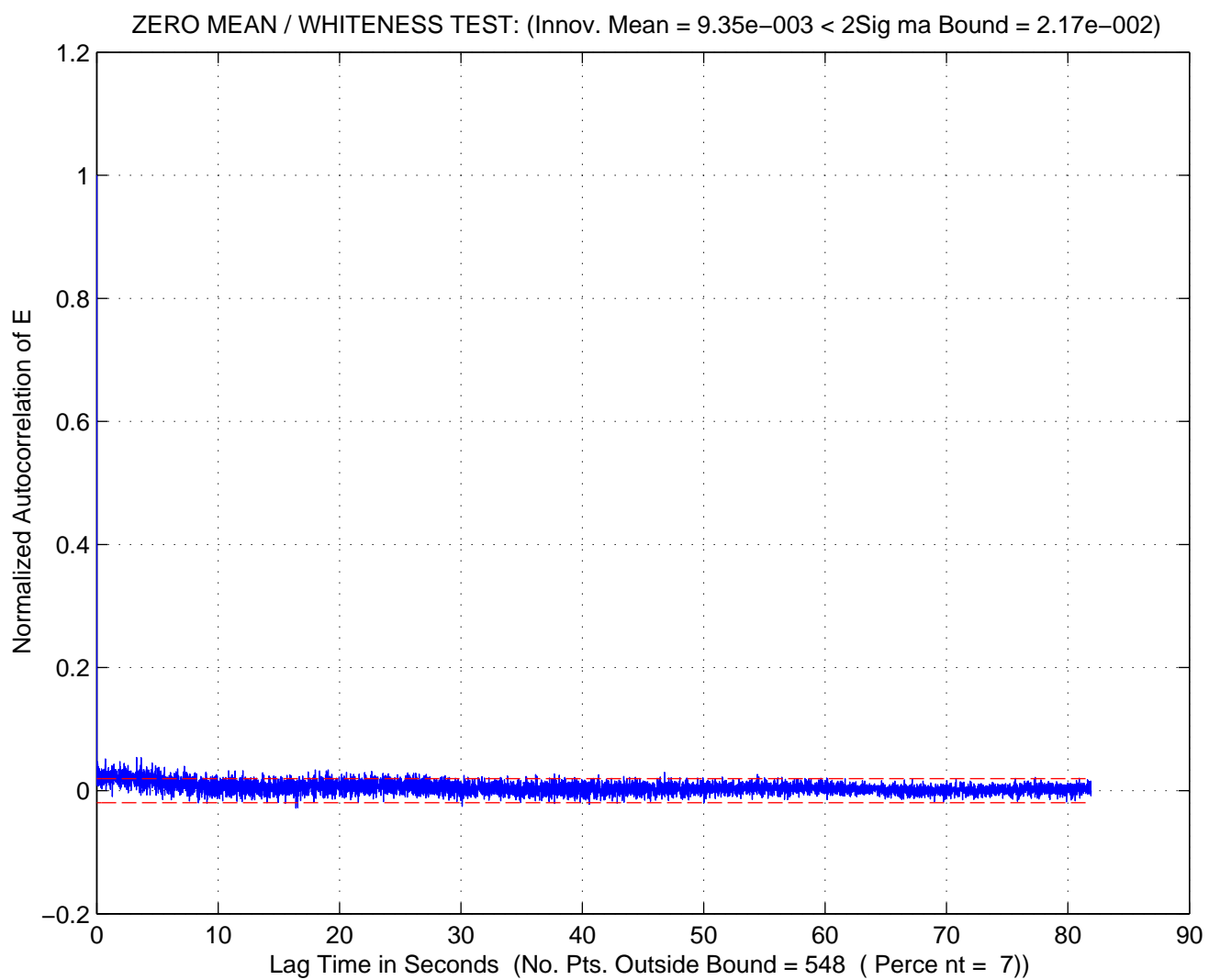


Figure 5.13: (Exp.1) Whiteness test for the y-component of the innovations  $e_y(k)$  (F45)



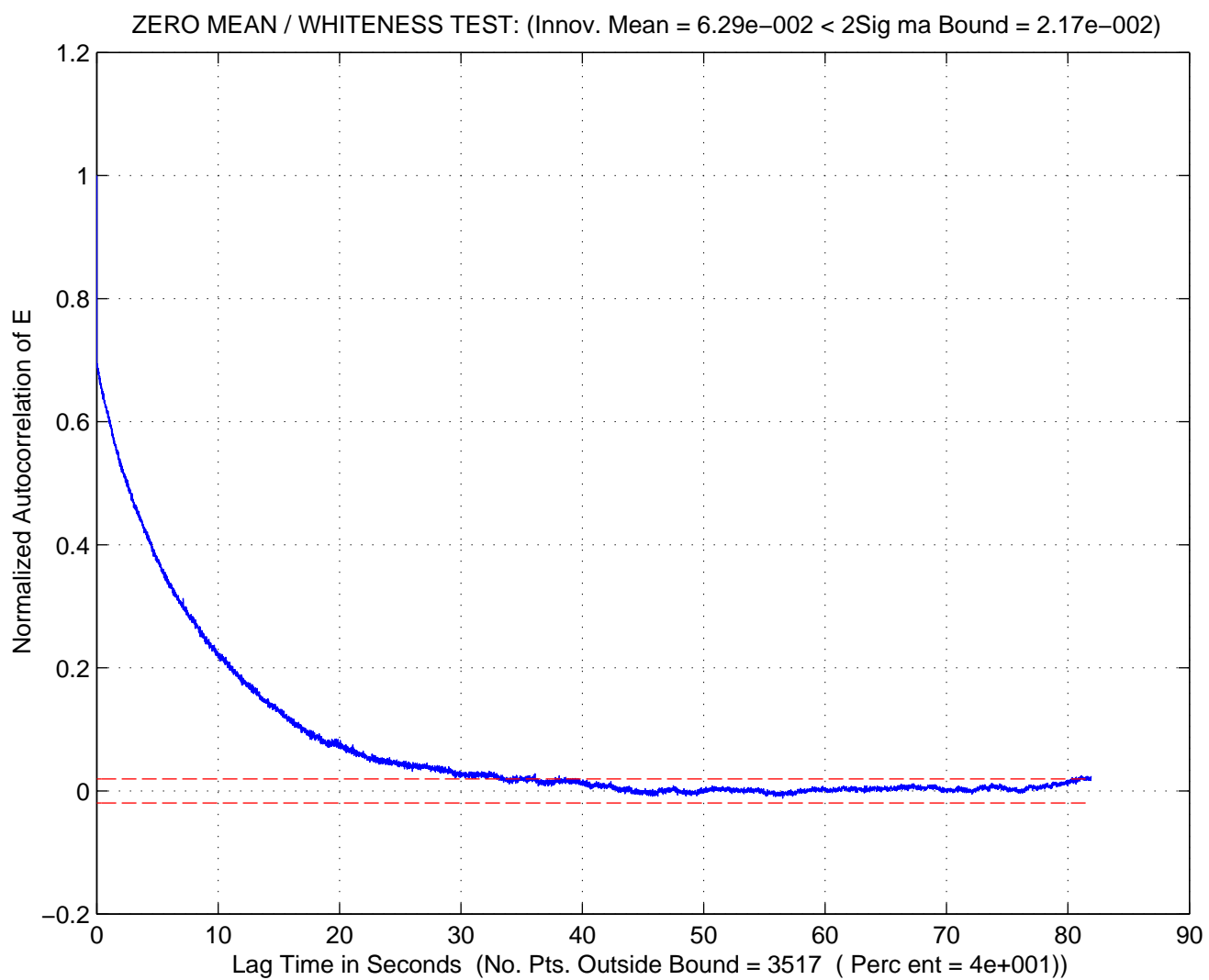


Figure 5.14: (Exp.1) Whiteness test for the z-component of the innovations  $e_z(k)$  (F46)

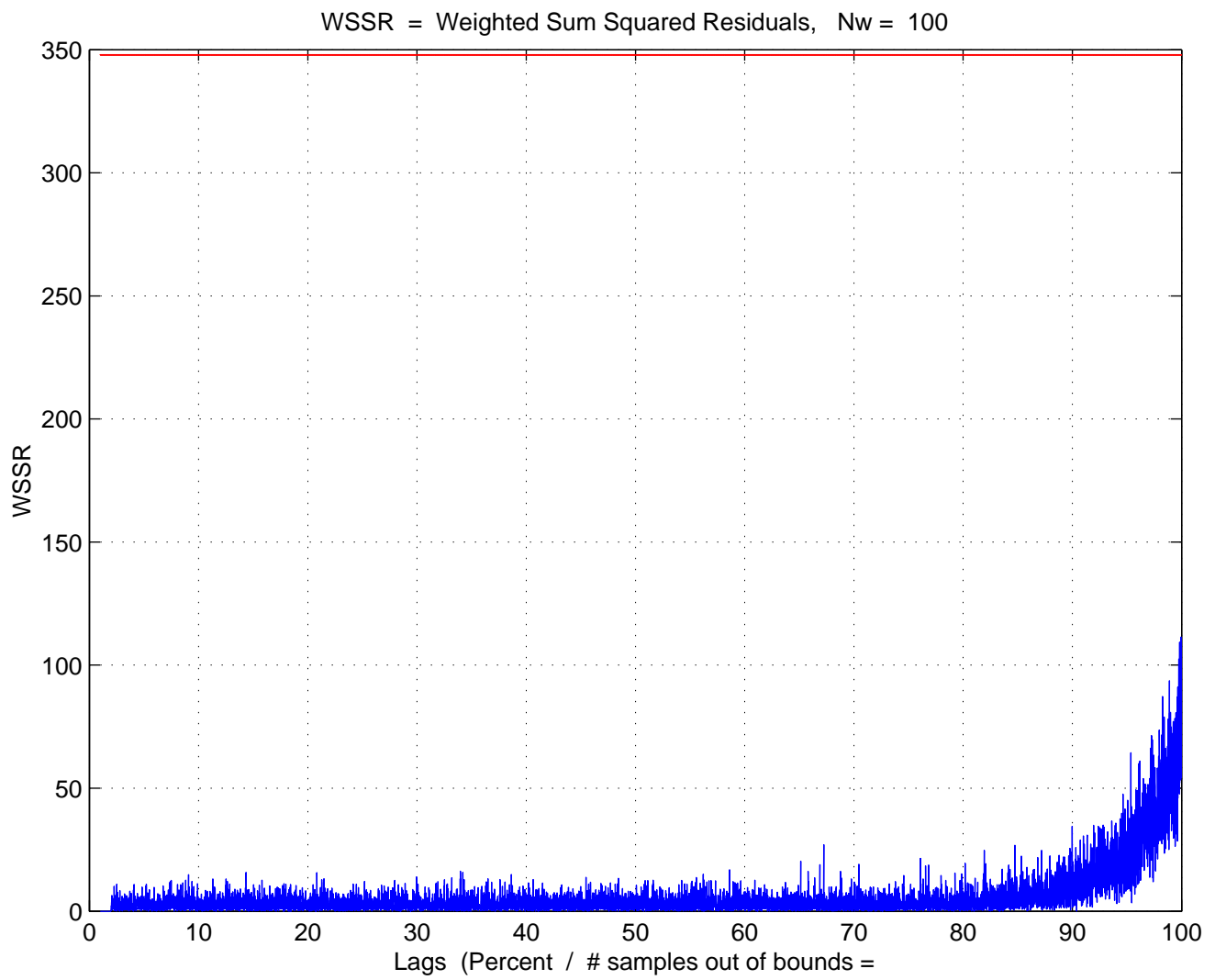


Figure 5.15: (Exp.1) Weighted sum squared residuals (WSSR) (F43)

$\underline{A}(k) - \underline{C}[\hat{\omega}(k)]$  when the quantities in this difference are large, but the difference is small. This is the case for this experiment. Figure (5.2) shows the accelerations  $\underline{A}(k)$ . Figure (5.16) shows the centripetal accelerations  $\underline{C}[\hat{\omega}(k)]$  from the EKF and  $\underline{C}[\omega(k)]$  from the ODE simulator for this experiment. The numerical problem is illustrated in Figure (5.17) which plots this key difference. We see that the difference is small compared with the values of  $\underline{A}(t)$  and  $\underline{C}[\hat{\omega}(k)]$ .

Quadrature to compute the linear velocities was not successful for this example. This is demonstrated in Figure (5.8). The numerical difficulty described above leads to this result.

The positive aspect of this analysis is that the Extended Kalman Filter adds great value in the presence of measurement noise. When noise is present, the classical ODE solver is not effective. However, the EKF is designed to handle noise, so the angular velocity and measurement estimates are of high quality (see Figure (5.7) and Figure (5.10)). In addition, the Weighted Sum Squared Residual (WSSR) meets the whiteness criterion, even though the solutions show drift at late times (see Figure (5.15)). This indicates the EKF is tuned sufficiently.

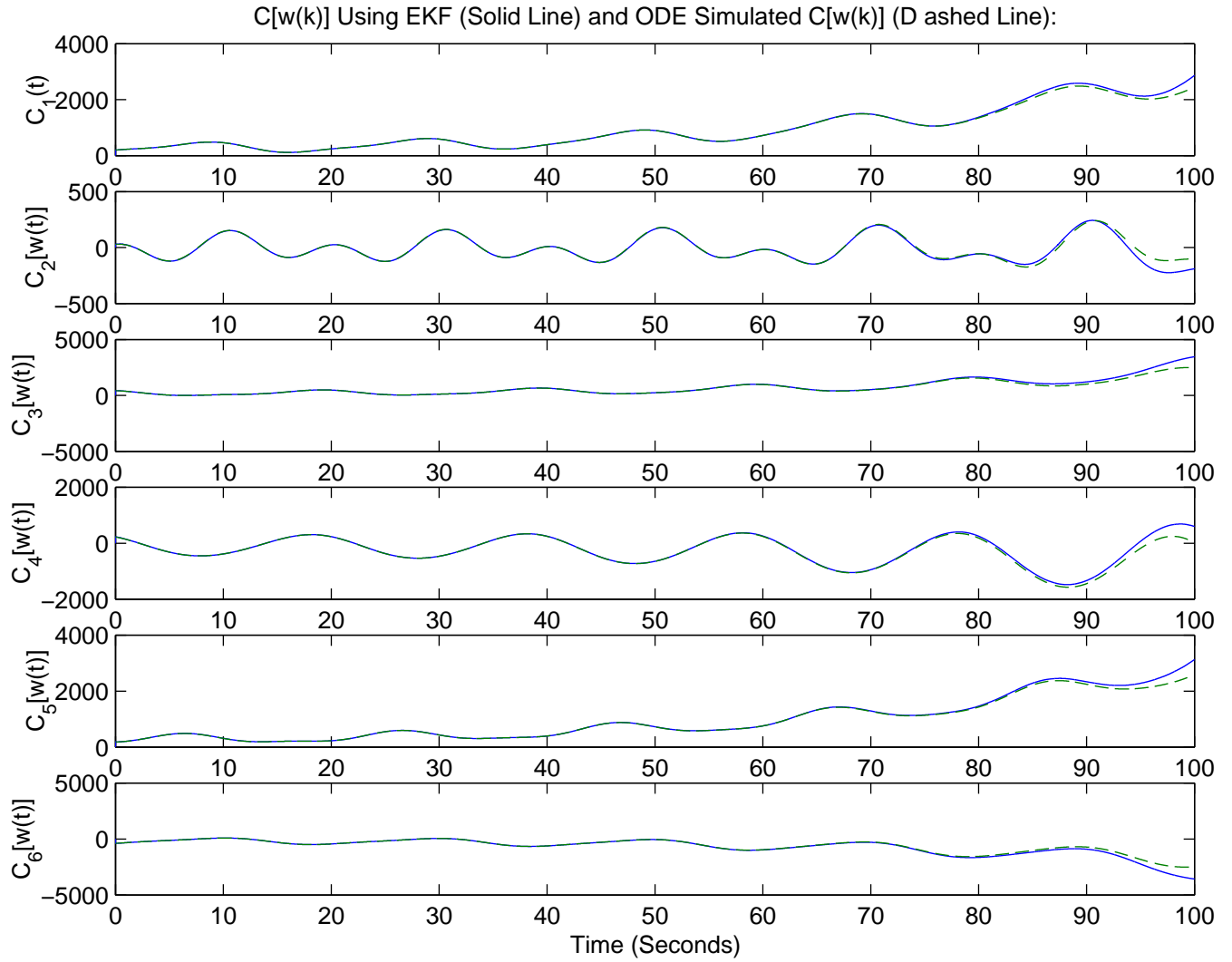


Figure 5.16: (Exp.1) Centripetal acceleration  $\underline{C}[\hat{w}(k)]$  from the EKF (solid) and  $\underline{C}[\underline{w}(k)]$  from the ODE simulator. Note that the accelerations are large. (F55)

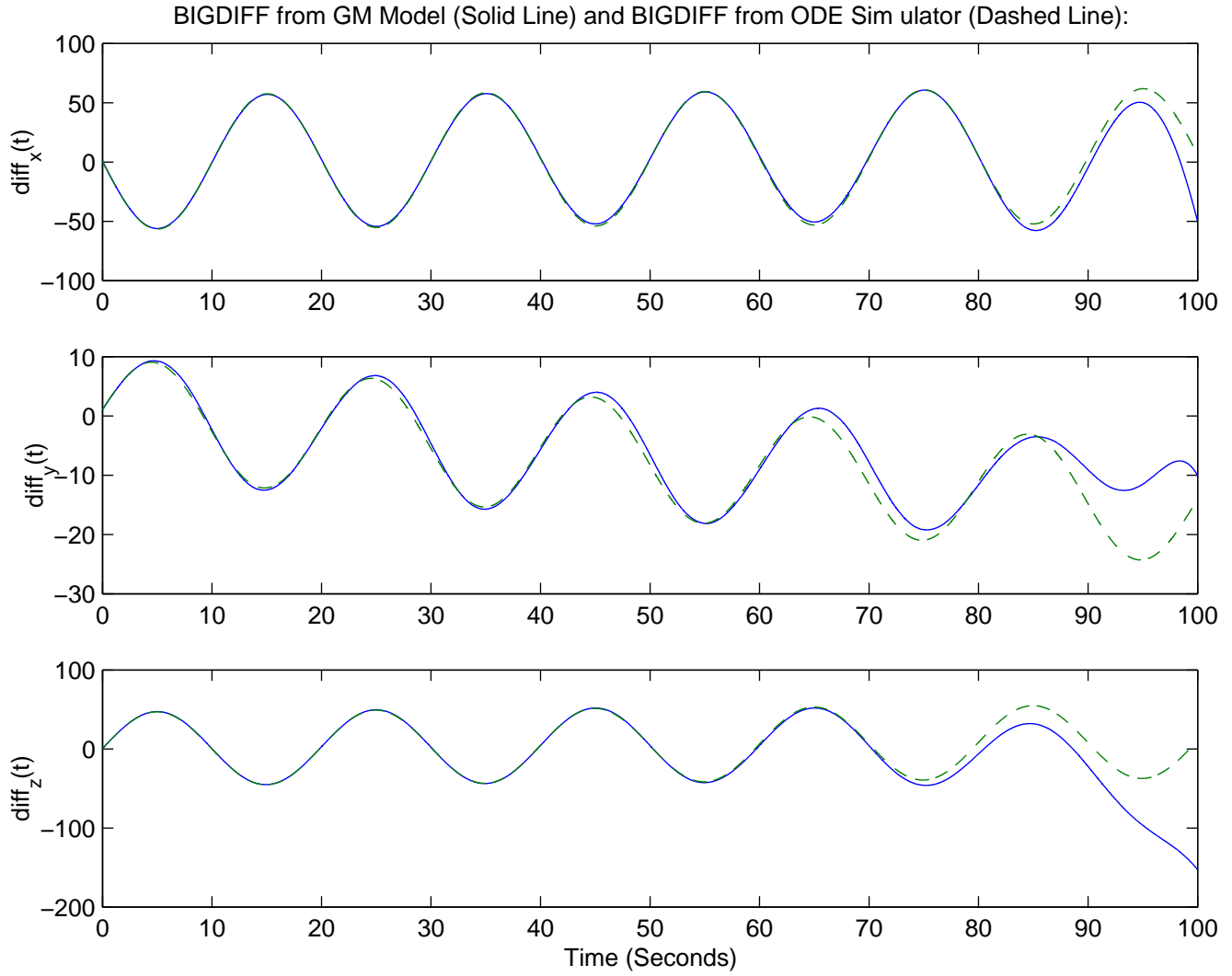


Figure 5.17: (Exp.1) Difference  $\underline{A}(k) - \underline{C}[\underline{\omega}(k)]$  from the Gauss-Markov model (solid) and from the ODE simulator (dashed). This shows the numerical difficulty associated with the ODE's and the example problem. (F36)

## Chapter 6

# Future Work

In addition to the mathematical/numerical issues described in the last section, the following issues are the subjects of current and future work.

### 6.1 Calibration and Error Compensation

An important issue under investigation is the fact that the actual sensor locations  $\underline{r}$  and directions  $\hat{\theta}$  have uncertainty associated with them. The tolerances associated with machining the DAIMU fixture can have a significant effect upon the velocity estimates. Carlos Avalle [9] has reported that the uncertainty in the accelerometer locations is on the order of a few mils.

These uncertainties have led to ongoing work to calibrate the DAIMU assembly using a rate table and other means. The plan is to test and calibrate single and multiple accelerometers. Part of this work involves building an error compensation scheme employing a parameter estimator for the accelerometer locations  $\underline{r}$  and directions  $\hat{\theta}$ .

### 6.2 Perturbation Analysis

Along with the calibration work, David Chambers [12] has performed sensitivity analyses showing that small perturbations in the accelerometer locations and directions can lead to significant errors in the velocity calculations, even when noise is not present. For example, The configuration matrix  $\tilde{T}$  (which must be inverted) is sensitive to these perturbations (See [12]). This work is ongoing.

### **6.3 Estimation Using Simulations of a Full Flight Profile**

David Chambers is building a simulator for the entire flight profile of interest for the re-entry vehicle. This includes separation, drift, spinup and re-entry. This simulator will allow us to analyze the estimator performance under realistic controlled conditions in preparation for the real data. This simulation should be more informative than the one used in this report.

### **6.4 Estimation Using Actual Flight Test Data**

Actual flight test data have been analyzed by David Chambers using an ODE solver [12]. The results indicate the need for state estimation, perturbation analysis, calibration work and simulation. Processing of these data with the state estimator will be the subject of a future report.

## Chapter 7

# Conclusions

We are interested in calculating the angular and linear velocities of a re-entry vehicle using six acceleration signals from a distributed accelerometer inertial measurement unit (DAIMU). Earlier work showed that angular and linear velocity calculation using classic ODE solvers is not practically feasible, due to mathematical and numerical difficulties.

In this report, we have performed an analysis that demonstrates the *theoretical feasibility* of using model-based nonlinear state estimation techniques to obtain the angular and linear velocities in this problem. Practical numerical and calibration issues require additional work to resolve.

We showed that the six accelerometers in the DAIMU are not sufficient to provide observability, so additional measurements of the states are required (e.g. from a Global Positioning System (GPS) unit). Given the constraint that our system cannot use GPS, we propose using the existing on-board 3-axis magnetometer to measure angular velocity. We further showed that the six ODE's for the vehicle kinematics can be decoupled into three ODE's in the angular velocity and three ODE's in the linear velocity. This allows us to formulate a three-state Gauss-Markov system model for the angular velocities, using the magnetometer signals in the measurement model. This re-formulated model is observable, allowing us to build an Extended Kalman Filter (EKF) for estimating the angular velocities. Given the angular velocity estimates from the EKF, the three ODE's for the linear velocity become algebraic, and the linear velocity can be calculated by numerical integration. Thus, we do not need direct measurements of the linear velocity to provide observability, and the technique is mathematically feasible.

In a simulation example, the estimator adds value over the numerical ODE solver in the presence of measurement noise. Calculating the velocities in the presence of significant measurement noise is not feasible with a classic ODE



solver. The EKF is able to deal effectively with the noise and provide useful angular velocity estimates. The linear velocity estimates for this simulation show numerical difficulties associated with the nonlinear ODE's and the quadrature operation.

Future work will focus on dealing with practical numerical issues and the issue of calibrating the DAIMU to deal with uncertainties in the accelerometer positions and locations.

# Bibliography

- [1] J-H Chen, S-C Lee and D. B. DeBra, *Gyroscope Free Strapdown Inertial Measurement Unit by Six Linear Accelerometers*, J. Guidance Control and Dynamics, 17(2), 1993, pp. 286-290.
- [2] A. R. Schuler, A. Grammatikos, K. A. Fegley, Measuring Rotational Motion with Linear Accelerometers, IEEE Trans. Aerospace and Electronic Systems, Vol. AES-3, No. 3, May, 1967, pp. 465-471.
- [3] A. J. Padgaonkar, K. W. Krieger, A. I. King, *Measurement of Angular Acceleration of a Rigid Body Using Linear Accelerometers*, Trans. ASME, September, 1975, pp. 552-556.
- [4] J. T. Gillis, *Estimation of 3-D Angular Motion Using Gyroscopes and Linear Accelerometers*, IEEE Trans. Aerospace and Electronic Systems, Vol. 27, No. 6, November 1991, pp. 910-920.
- [5] N. K. Mital and A. I. King, *Computation of Rigid-Body Rotation in Three-Dimensional Space from Body-Fixed Linear Acceleration Measurements*, Journal of Applied Mechanics, December 1979, Vol. 46, pp. 925-930.
- [6] J. Genin, J. Hong, W. Xu, *Accelerometer Placement for Angular Velocity Determination*, Trans. ASME, Vol. 119, September, 1997, pp. 474-477.
- [7] M. L. Psiaki and Y. Oshman, *Spacecraft Attitude Rate Estimation from Geomagnetic Field Measurements*, Journal of Guidance, Control and Dynamics, Vol. 26, Mar-Apr 2003.
- [8] C. A. Avale and J.I. Castor, *Distributed Sensor Inertial Measurement Unit*, Lawrence Livermore National Laboratory Report UCRL-53868-98, Engineering Research, Development and Technology FY'98, pp. 6:23-6:25.
- [9] C. A. Avale, *Design and Implementation of DAIMU*, Signal and Imaging Sciences Workshop, Center for Advanced Signal and Imaging Sciences, Lawrence Livermore National Laboratory, November 14-15, 2002.

- [10] R. J. Kane, *Overview of Data Analysis for Flight Tests*, Lawrence Livermore National Laboratory Report UCRL-PRES-150935, Signal and Imaging Sciences Workshop, Center for Advanced Signal and Imaging Sciences, Lawrence Livermore National Laboratory, November 14-15, 2002.
- [11] P. W. Kasameyer and L.J. Hutchings, *Mechanical Forces and Responses of a Re-Entry Vehicle Determined from Spectral Analysis of On-Board Data*, Signal and Imaging Sciences Workshop, Center for Advanced Signal and Imaging Sciences, Lawrence Livermore National Laboratory, November 14-15, 2002.
- [12] D. H. Chambers, *DAIMU Data Analysis - Performance and Stability*, Lawrence Livermore National Laboratory Report UCRL-PRES-150896, Signal and Imaging Sciences Workshop, Center for Advanced Signal and Imaging Sciences, Lawrence Livermore National Laboratory, November 14-15, 2002.
- [13] G. A. Clark, *A Model-Based Nonlinear Estimation Approach to DAIMU Signal Analysis for Re-Entry Vehicle Kinematics*, Lawrence Livermore National Laboratory Report UCRL-PRES-150726, Signal and Imaging Sciences Workshop, Center for Advanced Signal and Imaging Sciences, Lawrence Livermore National Laboratory, November 14-15, 2002.
- [14] G. A. Clark, *Re-Entry Vehicle Angular and Linear Velocity Estimation from DAIMU Signals*, Lawrence Livermore National Laboratory Report UCRL-PRES-152540, First Annual Sensors Workshop, Lawrence Livermore National Laboratory, April 2-3, 2003.
- [15] S. H. Crandall, D. C. Karnopp, E. F. Kurtz, D. C. Pridmore-Brown, *Dynamics of Mechanical and Electromechanical Systems*, McGraw-Hill, 1968.
- [16] H. Goldstein, *Classical Mechanics*, Addison-Wesley, 1980.
- [17] J. V. Candy, *Signal Processing: The Model-Based Approach*, McGraw-Hill, New York, NY, 1986.
- [18] J. V. Candy, *Signal Processing: The Modern Approach*, McGraw-Hill, New York, NY, 1988.
- [19] J. V. Candy, *On-Line Structural Parameter Estimation Using the Extended Kalman Identifier*, Lawrence Livermore National Laboratory report UCRL-83254, Pre-Print to the Journal of Sound and Vibration, August 21, 1979.
- [20] A. Gelb (Editor), The Technical Staff of the Analytic Sciences Corporation, *Applied Optimal Estimation*, the M.I.T. Press, 1974.
- [21] H. W. Sorenson, *Parameter Estimation*, Marcel Decker, Inc., 1980.

- [22] P. M. DeRusso, R. J. Roy, C. M. Close, *State Variables for Engineers*, John Wiley and Sons, Inc., 1965.
- [23] T. Kailath, *Lectures on Linear Least-Squares Estimation*, Springer-Verlag, 1976.
- [24] A. P. Sage and J. L. Melsa, *Estimation Theory with Applications to Communications and Control*, McGraw-Hill, 1971.
- [25] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, 1965.
- [26] T. Kailath, *Linear Systems*, Prentice-Hall, 1980.
- [27] Y. Takahashi, M. J. Rabins and D. M. Auslander, *Control and Dynamic Systems*, Addison-Wesley, 1972.
- [28] C.-T. Chen, *Introduction to Linear System Theory*, Holt, Rinehart and Winston, 1970.
- [29] A. Jazwinski, *Stochastic Processes and Filtering Theory*, Academic Press, New York, NY, 1970.
- [30] P. Maybeck, *Stochastic Models, Estimation, and Control*, Academic Press, New York, NY, 1979.
- [31] B.D.O. Anderson and J. B. Moore, *Optimal Filtering*, Prentice Hall, 1979.
- [32] R.K. Mehra and J. Peschon, "An Innovations Approach to Fault Detection and Diagnosis in Dynamic Systems", *Automatica*, Vol. 7, 1971.
- [33] F. Schweppe, *Uncertain Dynamic Systems*, Prentice-Hall, Englewood Cliffs, N.J., 1973.
- [34] J. V. Candy and R. B. Rozsa, *Safeguards design for a plutonium concentrator - an applied estimation approach*, *Automatica*, Vol. 16, pp. 615-627, 1980.
- [35] B. Noble, *Applied Linear Algebra*, Prentice-Hall, 1969.
- [36] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1989.
- [37] W. H. Press, B. P. Flannery, S. A. Teukolsky, W.T. Vetterling, *Numerical Recipes*, Cambridge University Press, 1987.
- [38] *MATLAB Reference Manual*, The Mathworks, Natick Massachusetts, 1993.

University of California  
Lawrence Livermore National Laboratory  
Technical Information Department  
Livermore, CA 94551

